UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Enhancing Robustness of Multilingual Transformer Feature Spaces

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Álvaro Huertas García
Master's Degree in Bionformatics and Computational Biology

Madrid, 2024

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Doctoral Degree in Computer Sciences and Technologies for Smart Cities

Enhancing Robustness of Multilingual Transformer Feature Spaces

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Álvaro Huertas García
Master's Degree in Bionformatics and Computational Biology

Under the supervision of:
Dr. David Camacho Fernández (Supervisor)
Dr. Alejandro Martín García (Co-supervisor)

Madrid, 2024

Title: Enhancing Robustness of Multilingual Transformer Feature Spaces

Author: Álvaro Huertas García

Doctoral Programme: Computer Sciences and Technologies for Smart Cities

Thesis Supervision:

Dr. David Camacho Fernández, Catedrático de Universidad, Universidad Politécnica de Madrid (Supervisor)

Dr. Alejandro Martín García, Profesor Titular, Universidad Politécnica de Madrid (Co-Supervisor)

External Reviewers:

Thesis Defense Committee:

Thesis Defense Date:

This work has supported by the research project CIVIC: Intelligent characterisation of the veracity of the information related to COVID-19, granted by BBVA FOUNDATION GRANTS FOR SCIENTIFIC RESEARCH TEAMS SARS-CoV-2 and COVID-19, by the project PCI2022-134990-2 (MARTINI) of the CHISTERA IV Cofund 2021 program, funded by MCIN/AEI/10.13039/ 501100011033 and by the "European Union NextGenerationEU/PRTR"; by the Spanish Ministry of Science and Innovation under FightDIS (PID2020-117263GB-I00); by MCIN/AEI/10.13039/501100011033/ and European Union NextGenerationEU/PRTR for XAI-Disinfodemics (PLEC2021-007681) grant, by European Comission under IBERIFIER Plus - Iberian Digital Media Observatory (DIGITAL-2023-DEPLOY-04-EDMO-HUBS 101158511); by "Convenio Plurianual with the Universidad Politécnica de Madrid in the actuation line of Programa de Excelencia para el Profesorado Universitario", by EMIF managed by the Calouste Gulbenkian Foundation, in the project MuseAI and by and by the research project "DisTrack: Tracking disinformation in Online Social Networks through Deep Natural Language Processing", granted by Barcelona Mobile World Capital Foundation.

Learning Without Thought Is Pointless.
Thought Without Learning Is Dangerous
- Confucio

Acknowledgement

"Aerodynamically, a bee's body is not made to fly; the good thing is that the bee doesn't know."

This interesting quote comes from a NASA poster. It talks about a time when scientists didn't understand how bumblebees could fly. Scientists thought that bees' wings were too small for their big bodies, so they shouldn't be able to fly.

Of course, we now know that bees actually can fly. Today, we understand the complex way their wings work, but this quote is still powerful as a metaphor for human potential.

Just as a bee flies despite apparent limitations, we can overcome obstacles that seem impossible. Regardless of the challenges we face or the doubts others may express, we have the power to persevere and succeed.

Let's embody the spirit of the bee, embracing life's sweetness and choosing to rise above any difficulty, without being intimidated by our perceived limitations.

Abstract

In the digital age, the proliferation of misinformation poses a significant threat to public discourse and societal well-being. This thesis, comprising three interconnected articles, addresses the critical challenge of combating false information through advanced Natural Language Processing (NLP) techniques. It focuses on enhancing the robustness and effectiveness of multilingual Transformer models in detecting and mitigating various forms of misinformation, with particular emphasis on optimizing feature spaces and maintaining semantic integrity across languages.

This research is motivated by the urgent need for sophisticated, adaptable, and globally applicable solutions to counter the spread of false information across linguistic and cultural boundaries. As the state-of-the-art review demonstrates, current trends in NLP highlight the importance of semantic understanding, multilingual capabilities, and the power of Transformer architectures. These trends, combined with the growing complexity of misinformation tactics, justify the focus and approach of this thesis.

The primary objectives were to develop innovative techniques to optimize feature spaces in multilingual Transformer models, create novel methods for detecting and counteracting content evasion techniques, explore adversarial training strategies, demonstrate the efficacy of multilingual approaches, investigate dimensionality reduction techniques, and identify solutions to critical computational challenges in combating misinformation.

The methodology encompassed three main areas of study. First, it explored feature space dimensionality reduction, maintaining semantic integrity across languages to support semi-automated fact-checking. Second, it developed a customizable simulation and generation text camouflage tool, focusing on identification through Name Entity Recognition and demonstrating the superiority of multilingual models in detecting evasion. Third, it assessed the impact of content evasion, compared countermeasures, and focused on in-model awareness to enhance resilience.

The research yielded several key innovations, including the pyleetspeak tool for simulating and detecting word camouflage, exploring adversarial training strategies, and applying Independent Component Analysis (ICA) for dimensionality reduction. These advancements significantly improved model performance and efficiency, particularly in multilingual contexts. The findings have substantial implications for NLP and misinformation combat efforts. The semantically-aware approach to dimensionality reduction enhances both the efficiency and effectiveness of misinformation detection systems. The development of tools for detecting camouflaged content addresses evolving tactics used by malicious actors. Moreover, the research consistently demonstrates the superiority of multilingual approaches in addressing the transnational nature of false information.

In conclusion, this thesis contributes to exploring new strategies for combating misinformation across linguistic boundaries. While acknowledging limitations such as the rapid pace of NLP advancements and real-world complexities, this work stands as a foundation for continued innovation.

Future research directions include exploring multimodal approaches, investigating ethical implications of AI-driven content moderation, expanding multilingual datasets, and further exploring advanced dimensionality reduction techniques in various NLP tasks.

In the evolving landscape of artificial intelligence and combating misinformation, this research underscores the crucial role of advanced NLP techniques, particularly Transformers and semantically-aware multilingual models, in promoting a more trustworthy digital landscape. This thesis emphasizes the need for ongoing vigilance, adaptability, and commitment to the ethical application of AI in service of truth and informed public discourse.

Resumen

En la era digital, la proliferación de la desinformación supone una importante amenaza para el discurso público y el bienestar de la sociedad.

Esta tesis consta de tres artículos interconectados que abordan el reto de combatir la problemática de la información falsa mediante Procesamiento del Lenguaje Natural (PLN). El estudio se enfoca en mejorar la robustez y eficacia de modelos multilingües Transformer para detectar y mitigar diversas formas de desinformación, optimizando el espacio de características y manteniendo la integridad semántica en múltiples idiomas.

La investigación responde a la urgente necesidad de soluciones avanzadas, adaptables y globales para contrarrestar la propagación de información falsa independiente del idioma. El estado del arte muestra la importancia de la comprensión semántica, las capacidades multilingües y el gran potencial de las arquitecturas Transformer, que junto con la creciente complejidad y constante evolución de las tácticas de desinformación, justifican el enfoque de esta tesis.

Los objetivos principales son explorar técnicas para optimizar los espacios de características en modelos Transformer multilingües, investigar técnicas de reducción de la dimensionalidad, crear métodos novedosos para detectar y contrarrestar las técnicas de evasión de contenidos, explorar estrategias de entrenamiento adversarial para aumentar la robustez frente a usuarios maliciosos, y demostrar la eficacia de los enfoques multilingües en la lucha contra la desinformación.

En primer lugar, se explora la reducción de la dimensionalidad del espacio de características, manteniendo la integridad semántica entre idiomas para ser empleado en la comprobación semiautomatizada de hechos. En segundo lugar, se desarrolla una herramienta personalizable de simulación y generación de camuflaje de textos, centrándose en la identificación mediante el reconocimiento de entidades camufladas y demostrando la superioridad de los modelos multilingües en la detección de la evasión. En tercer lugar, se evalua el impacto de la evasión de contenidos destacando el potencial del espacio de características de los modelos Transformers para mejorar la resistencia ante instancias camufladas.

La investigación aporta varias contribuciones clave, como la herramienta pyleetspeak para simular y detectar el camuflaje de palabras, la exploración y recomednación de estrategias de entrenamiento adversariales y la aplicación del Análisis de Componentes Independientes (ICA) para reducir la dimensionalidad y mejorar el rendimiento de los modelos. El desarrollo de herramientas para detectar contenidos camuflados aborda la evolución de las tácticas empleadas por los agentes maliciosos. Además, la investigación demuestra sistemáticamente la superioridad de los enfoques multilingües sobre los monolingües a la hora de abordar la naturaleza internacional de la información falsa.

A modo de conclusión, esta tesis contribuye a explorar nuevas estrategias para combatir la desinformación independientemente del idioma. No obstante, se reconocen limitaciones como el acelerado ritmo de avances en PNL y las complejidades de la aplicación en el mundo real.

Futuras líneas de investigación incluyen la exploración de enfoques multimodales, la investi-

gación de las implicaciones éticas de la moderación de contenidos por la IA, la ampliación de la cantidad de datos multilingües y la exploración de técnicas avanzadas de reducción del tamaño de los modelos.

En el dinámico panorama de la inteligencia artificial y la lucha contra la desinformación, esta investigación demuestra el papel crucial de las técnicas avanzadas de PNL, en particular los Transformers y los modelos multilingües con conocimiento de la semántica, para promover un entorno digital más fiable y sano. Esta tesis señala la necesidad de una vigilancia continua, adaptabilidad y compromiso con la aplicación ética de la IA al servicio de la transparencia y promoción del espíritu crítico.

Contents

List of Figures

List of Tables

Abbreviations

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

BoW Bag of Words

BPE Byte Pair Encoding

BPTT Backpropagation Through Time

CBOW Continuous Bag of Words

CD Contrastive Divergence

CLIP Contrastive Language–Image Pre-training

CLM Causal Language Modeling

CNN Convolutional Neural Network

CRF Conditional Random Field

C-RTS Cluster-based Random Token Substitution

DBN Deep Belief Neural Network

DL Deep Learning

DNN Deep Neural Network

EFR Error Finding Rate

FFNN Feed Forward Neural Network

FN False Negative

FP False Positive

GIFCT Global Internet Forum to Counter Terrorism

GMM Gaussian Mixture Model

GPT Generative Pretrained Transformer

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

HMM Hidden Markov Models

ICA Independent Component Analysis

IFCN International Fact-Checking Network

IPCA Incremental Principal Component Analysis

IR Information Retrieval

KPCA Kernel Principal Components Analysis

LaBSE Language-agnostic BERT Sentence Embedding

LLM Large Language Model

LSTM Long-Term Short-Term Memory

MAMI Multimedia Automatic Misogyny Identification

MCC Matthews Correlation Coefficient

ML Machine Learning

MLM Masked Language Modeling

MLQA MultiLingual Question Answering

MSE Mean Squared Error

NER Named Entity Recognition

NLI Natural Language Inference

NLP Natural Language Processing

NSP Next Sentence Prediction

OOV Out Of Vocabulary

OSN Online Social Network

PCA Principal Component Analysis

PD Performance Decrease

PLM Pre-trained Language Model

RBF Radial Basis Function Kernel

RBM Restricted Boltzmann Machine

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

RoBERTA Robustly Optimized BERT Approach

RoPE Rotatory Positional Encoding

RTS Random Token Substitution

SLM Swapped Language Model

SOP Sentence Order Prediction

SSL Self Supervised Learning

STS Semantic Textual Similarity

STSB Semantic Textual Similarity Benchmark

SVM Support Vector Machine

TAT Tech Against Terrorims

TF-IDF Term Frequency - Inverse Document Frequency

TLM Translation Language Modeling

TN True Negative

TP True Positive

UMAP Uniform Manifold Approximation and Projection

XLM Cross-lingual Language Model

XNLI Cross-Lingual Natural Language Inference

Chapter 1

Introduction

This thesis explores the rapidly evolving field of Artificial Intelligence, with a particular emphasis on the advancements in Natural Language Processing (NLP). It highlights the significant progress made in this field, underscoring the interest and development of sophisticated techniques, particularly in multilingual Transformers. The motivation behind this research originates from the need to deepen our understanding of these cutting-edge technologies, explore their intricate architectures, and enhance their applicability in addressing globally relevant challenges. Among the diverse potential applications, this thesis focuses on the critical and contemporary issue of information disorders, including misinformation and false information, particularly prevalent on social networks. The core objective is to explore and enhance the robustness of feature spaces in multilingual Transformers, specifically tailored to combat these types of information disorders to contribute to a cleaner and more trustworthy online environment.

This chapter presents an overview of the topics addressed in this thesis and its main contributions and motivations. It briefly introduces the problem while reserving technical details for later chapters since, in Chapter 2, there will be a precise exposition of the state-of-the-art as well as a formal definition of the problem.

The chapter starts by presenting the impact of the Artificial Intelligence revolution and the pivotal role of Natural Language Processing (NLP) in modern technology (Section 1.1). Following this, it defines the problem of false information, its impact on society and explains the need for computational solutions to address it (Section 1.2). After defining the problem, the chapter explains the motivations behind this thesis (Section 1.3) and the research objectives (Section 1.4). Then Section 1.5 presents the main contributions and publications generated during the realization of this thesis. Finally, the structure of this document is outlined (Section 1.6).

1.1 The Rise of Artificial Intelligence and NLP

The area of Artificial Intelligence (AI) has experienced a significant growth in recent years, which has impacted a huge number of academic fields. This growth is reflected in the

exponential increase in AI publications, which has more than doubled from 2010 to 2021, reaching almost 500,000, according to the Artificial Intelligence Index Report 2023 [1]. Over time, the source of groundbreaking machine learning models has shifted from academia to industry.

One of the most transformative developments is the advent of Deep Learning (DL), which has enabled AI systems to analyze and interpret complex data with unprecedented accuracy. AI significantly contribute to the growth of IoT applications in these areas, enhancing the functionality and efficiency of smart city infrastructures [2] and developing more efficient renewable energy systems [3]. In healthcare, AI algorithms assist in diagnosing diseases, predicting patient outcomes, and personalizing treatment plans [4]. These example advancements are significantly supported by substantial progress in machine learning algorithms, particularly neural networks, which have dramatically improved in processing and learning from large data sets [5].

Natural Language Processing (NLP), a critical subset of AI, has mainly influenced this AI revolution by introducing the Transformer architecture in 2017 [6]. This innovation revolutionized NLP by enabling models to process words concerning all other words in a sentence, rather than sequentially, significantly enhancing the understanding of context and meaning in language. The impact of this advancement is profound in academics, especially in linguistics research with the number of NLP publications reaching approximately 15,000 in 2021 [1], reflecting the field's rapid growth, and academic and industry interest.

However, with these advancements come the need for robustness in AI systems, to ensure they are reliable and secure [7]. Robustness in AI is critical to prevent errors and biases in decision-making, especially in sensitive scenarios like online social platforms [8]. The rapid increase in incidents concerning the misuse of AI underscores the need for advanced AI technologies like NLP Transformers to be robustly designed.

As AI continues to evolve, its impact on human language technology highlights the need for robust AI systems against malicious misuse [7]. The AI revolution, with NLP at its core, continues to reshape the technological landscape, suggesting a future where AI and human language technology are intricately intertwined and continuously advancing [9].

This continuous advancement in AI and NLP, particularly with multilingual Transformers, opens up new frontiers in addressing complex challenges such as the proliferation of false information, presented in the following section.

1.2 Application domain: False information and its proliferation

This section presents the well-known issue of false Information and its proliferation as a case study to assess the quality and effectiveness of computational developments proposed in this thesis. While the focus of this thesis remains firmly within the realm of Computational Science, the false information domain provides a rich and relevant context for evaluating the proposed solutions.

False information, commonly called "fake news", is a complex and varied problem that has been around for as long as verbal communication exists, but its relevance has been prominent for more than a century and remains so today [10], [11]. The term "fake news" has been widely used in recent years. However, it does not accurately capture the range of forms that false information can take, such as disinformation, misinformation, propaganda, unverified rumours, poor reporting, and messages containing hate or harmful information [11], [12].

The advent of social media platforms has only intensified this problem, as it allows for the rapid and widespread dissemination of false information [13]–[15]. This proliferation of false information constitutes a significant challenge to individuals and society [10], [12]. The sheer volume of online information makes it difficult to distinguish between credible and unreliable sources. Even official fact-checking services are overwhelmed by the massive content volume they must assess [12].

Furthermore, the spread of false information and misinformation since the COVID-19 pandemic has highlighted the urgent need for practical solutions to combat this problem. The rapid spread of misinformation about the virus and its origins, as well as false cures and preventive measures, has led to confusion and mistrust among the public [14], [16]. Moreover, this epidemic has also affected the ability of health officials and organizations to effectively communicate accurate information and guidelines to the public, further aggravating the problem [17], [18].

A recent Ipsos-UNESCO study conducted in September 2023 provides critical insights into the evolving landscape of information dissemination and public trust [19]. The study reveals that social media has surpassed traditional media sources like print, radio, and even television, becoming a primary source of information for many. This shift towards social media for information, especially during election campaigns, is crucial given the public perception of widespread disinformation on these platforms. Additionally, the phenomenon of hate speech online has been identified as a significant issue, with 67% of internet users encountering it, predominantly targeting LGBT+ people and ethnic or racial minorities.

These examples illustrate the ongoing and evolving nature of the false information problem and the need for research and development of solutions to combat it [20]. In response to this crisis, there has been a notable increase in efforts to mitigate misinformation. Since 2018, the number of fact-checking sites has grown by 47%, demonstrating a growing awareness and proactive approach to the misinformation crisis. Alongside this, the increasing interest in this issue among scientists and the public has led to a significant increase in academic research aimed at finding solutions [12], [20].

However, despite these efforts, challenges and limitations still need to be addressed in effectively combatting the evolving nature of false information and the malicious actors that spread it. For instance, the effectiveness of fact-checking sites is often challenged by the sheer volume and complexity of false information being disseminated. Additionally, while making strides, the academic and scientific community still faces limitations in developing comprehensive solutions that can keep pace with the rapid spread and sophistication of misinformation tactics [11], [12].

Accordingly, there is a clear need for computational solutions to combat this problem. The

following section present the motivation of this thesis to propose innovative solutions that can help mitigate its impact.

1.3 Motivation

As mentioned above, the problem of false information, commonly known as "fake news", has become increasingly frequent in recent years. The ease of access to information through Online Social Networks (OSNs) has led to the proliferation of false information, making it difficult for people to distinguish between credible and unreliable sources [15], [17]. This situation provides an opportunity for malicious actors to take advantage of them [11]. A clear example was the previously mentioned COVID-19 pandemic [18], which has highlighted the urgent need to find practical solutions to combat the spread of misinformation [20], [21]. In later chapters of the thesis, we will further discuss the terminology and ecosystem of false information to better understand the problem.

One of the ways to combat false information is through fact-checking [12], [21]. Fact-checking is the practice of evaluating the truthfulness of claims appearing in public through reliable sources. While human fact-checking has been the traditional approach to addressing false information, the overwhelming amount of information present online makes it unfeasible to rely solely on manual methods [20], [21]. The overload of information in today's digital landscape makes traditional fact-checking methods, which rely heavily on manual labour, unfeasible to effectively combat the spread of false information. Even more, the massive volume of content, coupled with the speed at which it is disseminated, renders human fact-checking inadequate [14].

Due to this information overload, other solutions to cope with false information are also compromised. For example, human-driven solutions based on social corrections have been proposed as a promising approach to address the problem of false information [22]. Social corrections involve individuals correcting other users' posts in an attempt to combat the spread of misinformation. However, given the vast amount of false information online and how quickly it spreads, it is unrealistic and infeasible to rely solely on manual solutions for proper and efficient countermeasures [11], [20], [22]. Reliance on manual solutions alone must be improved to combat the issue effectively on the scale and with the urgency it demands. This highlights the need for new solutions that can assist human-supervised activities, such as automating bottlenecks and improving procedures, to keep pace with the rapidly evolving nature of false information.

In this context, computational science has emerged as a valuable tool to address this problem [11], [12], [20]. The rapid advancement of technology and the increasing availability of large-scale datasets provide the opportunity to develop sophisticated algorithms and models to detect and mitigate the spread of false information.

In light of this, the field of computer science offers a range of solutions to address this problem, from automating bottlenecks in the fact-checking process to improving and streamlining workflows [12], [16]. The application of computational techniques can aid by analyzing and processing large amounts of data and providing insights that would be otherwise impossible

to obtain manually. As such, it is imperative to leverage the capabilities of computer science to develop effective solutions that can keep pace with the rapidly evolving landscape of false information.

This thesis aims to contribute to the scientific debate by using computational solutions in Natural Language Processing and Deep Learning, specifically through the exploration of neural network architectures and their application in combating information disorders.

1.4 Goals, Thesis Outline and Research questions

This doctoral thesis addresses the critical issue of false information through the lens of Computer Science, leveraging advanced Deep Learning tools based on Natural Language Processing (NLP). The research focuses on harnessing the power of **Transformer** architecture, **feature space** optimization, **multilingual** capabilities, and **semantic awareness** to combat the spread of misinformation effectively.

1.4.1 General Objective

The overarching aim of this thesis is to develop and enhance computational methods for detecting and mitigating false information across multiple languages. This objective is pursued through the innovative application of NLP techniques, with a particular focus on optimizing the feature space of multilingual and semantically aware models built on the Transformer architecture.

1.4.2 Specific Objectives

To achieve the general objective, the following specific goals have been identified:

- **Dimensionality Reduction in Multilingual Semantics:** Investigate and implement dimensionality reduction techniques to enhance the efficiency and effectiveness of multilingual Transformer models in processing and analyzing text data.
- **Content Evasion Detection:** Develop robust methods for identifying and counteracting content evasion techniques, particularly focusing on word camouflage in multilingual contexts.
- **Enhancing Model Robustness:** Improve the resilience of Transformer models against adversarial attacks and evasion attempts through advanced training techniques and model adaptations.

1.4.3 Research Context and Significance

The Transformer architecture, introduced in 2017 [6], remains at the forefront of NLP advancements. Its adaptability and efficiency in handling large datasets make it an ideal foundation for addressing the complex challenges posed by false information. This thesis leverages the Transformer's capabilities while focusing on three key aspects, all unified by a multilingual perspective:

- **Feature Space Optimization:** By fine-tuning the feature space of Transformer models, we aim to enhance their ability to identify and interpret nuanced linguistic patterns crucial for detecting misinformation.
- **Semantic Awareness:** The incorporation of semantic knowledge allows our models to understand contextual meanings and word interactions, which is essential for accurately identifying and mitigating false information.
- **Content Evasion and Adversarial Robustness:** Addressing the growing challenge of content evasion techniques where malicious user try to alter the functionality of models in controlling a safe virtual environment, this research explores methods to detect camouflaged content and enhance model robustness against adversarial attacks.

Multilingual Perspective: Recognizing that false information transcends language barriers, this research adopts a multilingual approach throughout all aspects of the study. This overarching multilingual focus aims to overcome the limitations of monolingual solutions and address the global nature of misinformation spread, ensuring that our proposed solutions are applicable and effective across diverse linguistic landscapes.

The thesis will also address several specific objectives and research questions (RQ), which will be answered in Section 5.5. These are as follows:

1.4.4 Research Questions

Based on these objectives and context, the following research questions (RQ) guide this thesis, which will be answered in Section 5.5:

Principal Research Questions

- **RQ P1** - How can the dimensionality of feature spaces in multilingual transformer models be optimized to enhance semantic integrity and improve model performance in processing diverse linguistic data?
- **RQ P2** - What methods can be developed to detect and counteract content evasion techniques in multilingual contexts, and how effective are these methods in identifying camouflaged content within transformer models?
- **RQ P3** - How can the understanding of content evasion tactics be integrated into the feature space of NLP models, and what training strategies, such as adversarial training, are most effective in preparing these models to combat sophisticated misinformation techniques?

Additional Research Questions

- **RQ A1** - Can the fight against misinformation be approached from a multilingual point of view?
- **RQ A2** What is the performance impact of using the multilingual approach versus monolingual models? Consequently, is it worth extending monolingual datasets to a

multilingual level?

- **RQ A3** - Can dimensionality reduction techniques developed for multilingual transformer models be applied in a domain-agnostic framework to combat misinformation and its propagation?
- **RQ A4** - What are the bottlenecks and challenges in the complex and ongoing fight against misinformation and how can they be solved from a computational point of view?

1.5 Contributions and Publication record

This section presents the compilation of publications that form the basis of this thesis, with detailed descriptions of the quality indices and contributions made by the PhD candidate for each of them. The scientific journals A1, A2, and A3 comprise the collection of articles constituting this thesis by compendium and are presented in Chapter 4.

1.5.1 Publications of the compendium and Contributions

- P1: **Á. Huertas-García**, A. Martín, J. Huertas-Tato, y D. Camacho, «Exploring Dimensionality Reduction Techniques in Multilingual Transformers», Cognitive Computation, oct. 2022, doi: 10.1007/s12559-022-10066-8.
  - **Journal Ranking**: Cognitive Computation 2022: JCR 2022. Impact factor 5.4. Indexed in Neurosciences (Q1, 61/272) and Computer Science, Artificial Intelligence (Q2, 47/145).
  - **Summary**: The first research paper examines how different dimensional reduction techniques affect the performance of state-of-the-art **multilingual** Siamese transformers. The study focuses on the dimensionality of **feature spaces** while ensuring that semantics remains the core element for text meaning extraction. The results of this research significantly contribute towards developing an effective semi-automated fact-checking tool, which is detailed in the following paper. Additionally, this study highlights the potential of dimensional reduction techniques to deal with high-dimensional embeddings for other demanding NLP tasks.
  - **Contribution of the PhD candidate**:
    - First author of the article.
    - Conception and development of the research idea.
    - Design and implementation of the experiments.
    - Analysis and interpretation of results.
    - Primary author of the manuscript, including creation of visualizations.
- P2: **Á. Huertas-García**, A. Martín, J. Huertas-Tato, y D. Camacho, «Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflage», Applied Soft Computing, vol. 145, p. 110552, 2023, doi:

10.1016/j.asoc.2023.110552.

- **Journal Ranking**: Applied Soft Computing 2023: JCR 2023. Impact factor 7.2. Indexed in Computer Science, Interdisciplinary Applications (Q1, 15/169) and Computer Science, Artificial Intelligence (Q1, 27/197).
- **Summary**: The second paper suggests a new way to tackle content evasion on social networks by utilizing **multilingual** transformer models. The paper focuses on enhancing the **feature space** of these models to develop a Named Entity Recognition (NER) model that can identify disguised content, thereby improving the ability to detect subtle linguistic changes across different languages. This is a vital step in the fight against global misinformation. Furthermore, the paper introduces "pyleetspeak", a versatile multilingual tool that enables researchers to generate and simulate content evasion by disguising words. The tool has been designed for the scientific community to help combat the spread of misinformation.
- **Contribution of the PhD candidate**:
  - First author of the article.
  - Conceptualization of the approach to tackle content evasion.
  - Development and implementation of the "pyleetspeak" tool.
  - Design and execution of experiments using multilingual transformer models.
  - Lead author in writing the manuscript and creating visual representations of data.

- P3: **Á. Huertas-García**, A. Martín, J. Huertas-Tato, and D. Camacho, "Camouflage is all you need: evaluating and enhancing transformer models robustness against camouflage adversarial attacks," IEEE Trans. Emerg. Top. Comput. Intell., pp. 1–13, 2024, doi:10.1109/TETCI.2024.3440181.

  - **Journal Ranking**: IEEE Transactions on Emerging Topics in Computational Intelligence 2024. JCR 2023. Impact factor 5.3. Indexed in Computer Science, Artificial Intelligence (Q1, 43/197).
  - Summary: This study investigates the vulnerability of advanced language models to adversarial word camouflage techniques. It examines various model architectures (encoder-decoder, encoder-only, and decoder-only) and their tokenizers, revealing significant weaknesses against camouflaged words, particularly in keyword attacks. The research evaluates external countermeasures like MASK and BLANK filters and proposes novel static and dynamic adversarial training methods. The study also contributes to the field by enhancing the open-source tool pyleetspeak, facilitating the creation of augmented camouflaged datasets for strengthening NLP systems against evolving digital communication threats.
  - **Contribution of the PhD candidate**:

  - First author of the article.
  - Formulation of the research question and study design.
  - Implementation and evaluation of various language model architectures.
  - Development and application of static and dynamic adversarial training methods.
  - Primary contributor to the manuscript, including data analysis and discussion of results.

1.5.2 Other Publications and Contributions

Scientific Journals

- A4: A. Martín, J. Huertas-Tato, **Á. Huertas-García**, G. Villar-Rodríguez, and D. Camacho, «FacTeR-Check: Semi-automated fact-checking through semantic similarity and natural language inference», Knowledge-Based Systems, vol. 251, p. 109265, sep. 2022, doi: 10.1016/j.knosys.2022.109265.
  - Summary: FacTeR-Check is a multilingual architecture for semi-automated fact-checking and hoax propagation analysis that can be used to implement applications designed for both the general public and fact-checking organizations. This paper applies insights from the dimensionality research in the **feature space** of **multilingual** transformer models to develop the semi-fact-checking tool, natural language inference, and content search query building through automatic keyword extraction. It demonstrates practical use of the above feature space dimensionality reduction techniques for effective misinformation detection across languages.

International Congresses

- C1: **Á. Huertas-García**, A. Martín, J. Huertas-Tato, and D. Camacho. «Evaluación de modelos multilingües preentrenados en similitud semántica para la lucha contra la desinformación». In XIX Conference of the Spanish Association for Artificial Intelligence. Springer, 2021.
  - Summary: This paper demonstrates the usefulness of bi-encoder versus cross-encoder models and the ability of multilingual models to outperform monolingual models in semantic similarity tasks that require understanding the level of semantic similarity between a pair of sentences.
- C2: **Á. Huertas-García**, J. Huertas-Tato, A. Martín, and D. Camacho, «Countering Misinformation Through Semantic-Aware Multilingual Models», in Intelligent Data Engineering and Automated Learning – IDEAL 2021, 2021, pp. 312–323, doi: 10.1007/978-3-030-91608-4_31.
  - Summary: This work presents an approach to countering misinformation through a semantic-aware multilingual architecture. We built an extension of the well-known Semantic Textual Similarity Benchmark (STSb) to 15 languages. This new dataset

allows for fine-tuning and evaluating multilingual models based on Transformers with a siamese network topology. Results show that semantic fine-tuning improves measuring the degree of similarity between pairs of texts, while broadening our understanding of the multilingual capabilities of the models.

- C3: **Á. Huertas-García**, J. Huertas-Tato, A. Martín, and D. Camacho, «Profiling Hate Speech Spreaders on Twitter: Transformers and mixed pooling», CLEF (Working Notes), vol. 2021, 2021.
  - Summary: Authors Profiling Hate Speech Spreaders in Twitter Spanish and English tasks at CLEF 2021. The task consists in determining whether an author spreads hate speech given his/her Twitter feed. The proposed system uses semantic-aware Transformer-based models on tweets with mixed pooling for author profiling in combination with Sentiment Analysis and Hate lexicons to boost the feature extraction process.
  - Available in: https://pan.webis.de/downloads/publications/papers/huertasgarcia_2021.pdf
- C4: **Á. Huertas-García**, J. Huertas-Tato, A. Martín, and D. Camacho, «CIVIC-UPM at CheckThat! 2021: Integration of Transformers in Misinformation Detection and Topic Classification», in CLEF (Working Notes), 2021, pp. 520–530.
  - Summary: This work presents an NLP approach that uses Doc2Vec and different state-of-the-art transformer-based semantic-aware models for the CLEF2021 Checkthat! Task 3, which is divided into two subtasks. Task A is designed to classify a set of news into four classes (false, partially false, true, and other). Task B classifies a subset of news from Task A into six topical categories: health, economy, crime, climate, elections, and education.
  - Available in: https://ceur-ws.org/Vol-2936/paper-41.pdf
- C5: **Á. Huertas-García**, H. Liz, G. Villar-Rodríguez, A. Martín, J. Huertas-Tato, and D. Camacho, «AIDA-UPM at semeval-2022 task 5: Exploring multimodal late information fusion for multimedia automatic misogyny identification», in Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), 2022, pp. 771–779, doi: 10.18653/v1/2022.semeval-1.107.
  - Summary: This paper's main contribution is exploring different multimodal late fusion methods to increase the performance of a Transformer model for text in combination with Convolutional Neural Networks (CNN) for images proposed in the Multimedia Automatic Misogyny Identification (MAMI) task of SemEval-2022. Similarly, a novel methodology for pre-processing images in conjunction with text is presented. The presented results help address women's inequality and discrimination on social media platforms.
- C6: J. Huertas-Tato, A. Martin, **Á. Huertas-Garcia** and D. Camacho, «Generating Authorship Embeddings with Transformers», 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8, doi: 10.1109/IJCNN55064.202

2.9892173.

– Summary: This paper presents a methodology for authorship attribution and profiling tasks, which consists of finding a document's author according to different features that can be extracted from the text. These solutions can have broad fields of application, such as the detection of disinformation disseminators. Our approach can be used to extract representative authorship embeddings and visualize meaningful relationships between authors, genres, and books.

1.6 Organization of the Thesis manuscript

This thesis is structured into five chapters, which are summarized below:

- **Chapter 1 - Introduction**: An overview of the topic of the study, the motivations behind it, the contributions, and the research questions to be answered throughout the thesis.
- **Chapter 2 - Background and State-of-the-art**: This chapter provides a comprehensive overview of current literature and research in Natural Language Processing. It specifically focuses on the detection and mitigation strategies employed in combating the spread of false information through computational science. It also provides context for the research objectives of the thesis by highlighting the current challenges and limitations in the field.
- **Chapter 3 - Methodology**: This chapter outlines the methodological approach adopted in the study. It offers a detailed description of the development process and the evaluation techniques employed to assess the results. The chapter provides a comprehensive context of the approaches, methods, and data utilized to address the research questions outlined in Section 1.4, ensuring transparency and reproducibility of the research process.
- **Chapter 4 - Article Collection**: This chapter presents the curated collection of English articles that form the compendium thesis corpus.
- **Chapter 5 - General Discussion**: The final chapter offers an integrated description of the proposed solution, a critical analysis of the results, and draws overarching conclusions. It also explores potential future lines of research from this work. Importantly, this chapter elucidates the contributions of each publication included in Chapter 4, justifying how they collectively satisfy the research objectives and demonstrate the thematic unity of the solution and its results. This synthesis provides a holistic view of the research journey, its outcomes, and its implications for computational science in combating false information.

Chapter 2

Background

2.1 Literature Review on False Information

In the previous chapter, we talked about the increasing amount of false information, also known as "fake news", "misinformation", and "disinformation". This section aims to provide a thorough overview of the ongoing debate surrounding these terms and to explain the issue that this thesis addresses. After conducting a thorough analysis of the literature, it becomes evident that conventional approaches to tackling the issue of false information, like manual fact-checking, need to be reconsidered. This is because the vast amount of false information spread via the Internet is both rapid and widespread, necessitating computational solutions to complement existing methods.

2.1.1 Fundamentals of terminology and its evolution

The term “fake news” has become widely used in recent times, particularly among journalists and social media users, becoming a part of the common lexicon [12]. It refers to false information that one or more news sources have reported and, more broadly, to the use of information systems to manipulate people’s opinions [15]. However, this definition is not specific enough, as it does not include other entities that deliberately produce and spread false information. Moreover, it does not address social networks and large Internet platforms that act as global carriers of such misleading contentt [23].

The formal definition of “fake news” has been a topic of controversy among the scientific community, which has long discussed the correct definition of the term [11], [24], [25]. As a result, the earliest attempts reduced the definition of “fake news” to false information disseminated for harmful purposes [11], [12], [23]. Although it is accepted as a good definition because it includes both the veracity of the news and the malicious intent of the author, it is only part of a larger and more complex problem [24], [25].

Since 2017, a growing group of media experts and institutions, such as the Poynter Institute[1] the Council of Europe [25] and UNESCO [24], propose to stop using the term “fake news”,

[1] https://www.poynter.org/

as it is insufficient to represent the wider range of malicious activities and different ways in which false information online may come. Instead, they proposed that the "fake news" buzzword needs to be replaced by proper terminology that covers the veracity of the news and the malicious intentions of the author, considering that one term could not completely define all the diversity, but could include them all. Therefore, they proposed a more comprehensive definition of "information disorder", a term that, in turn, better describes the complexity of the false information ecosystem. This new accurate terminology of "information disorders" and false information covers the veracity of the news and the malicious intent of the author by encompassing other terms such as "misinformation", "disinformation", and "mal-information" problems that will be covered in the following Section.

2.1.2 Information disorders: misinformation, disinformation, and mal-information

We have already shown how experts consider the term "fake news" inadequate to explain the magnitude of information pollution and why it has become so problematic that we should avoid using it. We will now delve into alternative definitions considered more appropriate for addressing the issues associated with "information disorder".

According to the UNESCO Handbook [24] and the in-depth analysis carried out by Karlova and Fisher [26] we can establish the following definitions:

- ***Misinformation*** refers to false or inaccurate information that is spread unintentionally, often as a result of a lack of knowledge or understanding of a topic and usually the person who is disseminating it believes that it is true. An example of this would be sharing with a friend the efficacy of certain homemade treatments to grow hair without scientific evidence.
- ***Disinformation*** refers to false or misleading information that is deliberately spread with the intention of deceiving or manipulating. Hence, the person who is disseminating it knows that it is false and that it is an intentional lie and points to people being actively disinformed by malicious actors. An example of this would be false claims about the effectiveness of a hair growth product, without any scientific evidence, in order to increase sales and profit.
- ***Malinformation*** refers to genuine (partially) true information that is shared with malicious intent, such as the sharing of personal information to harm an individual or group. An example of this would be intentionally disseminating compromising personal information about a manager of a competing hair product in order to influence sales. Although those messages may contain some truth, but are created, produced, or distributed with the intention to harm rather than serve the public interest.

Overall, as the Figure 2.1 shows, there are two variables to consider when dealing with false information, the author's willingness to harm and the conveyed information's falsehood. However, many features can be considered, leading to a more extensive classification of information. It should also be noted that the consequences on the information environment and society may be similar and exhibit combinations of these three terms.

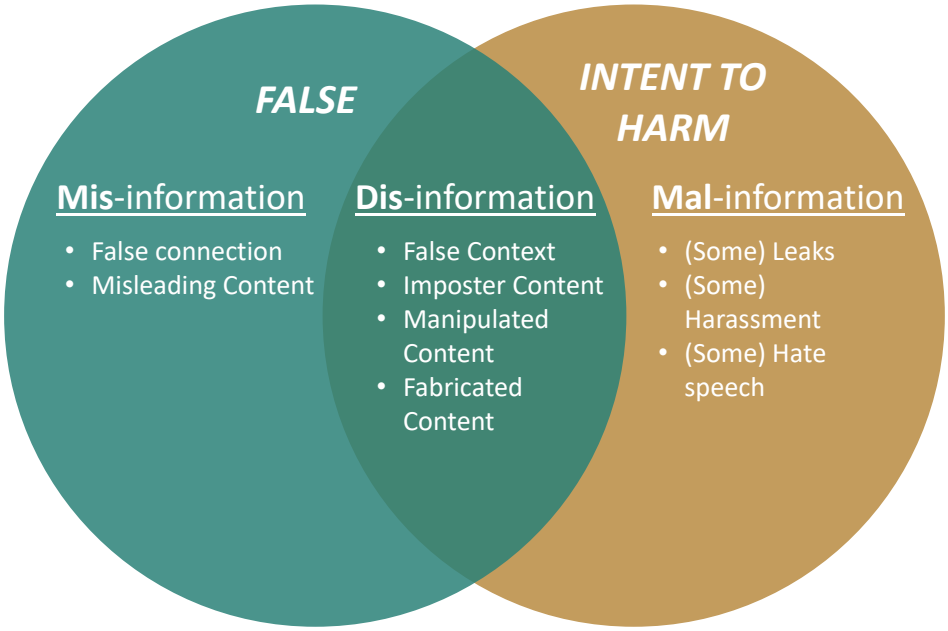


Figure 2.1: Graphical representation of the overlap between the concepts of misinformation, disinformation and malinformation belonging to the information disorder ecosystem. (Credit: Claire Wardle & Hossein Derakshan from firstdraftnews.org, 2017)

It is important to note that while the main concepts of disinformation, misinformation, and mal-information provide a general framework for understanding the problem of information disorder, there are additional terms and definitions that are relevant to the work presented in this thesis. These definitions have been included because they are considered important for the research presented and are necessary to provide a more complete understanding of the problem. Additional definitions have been extracted from [12] and are listed below:

- **Hoax**: disinformation that can also have humorous purposes [27].
- **Rumour**: unverified information that may or may not be true and accurate. If it is falsified, it is misinformation [28].
- **Infodemic**: mixture of misinformation and true information about the origins and alternative cures of a disease; especially observed during the COVID-19 pandemic [17], [18].
- **Hate speech**: abusive mal-information that targets certain groups of people, expressing prejudice and threatening [29].
- **Troll**: social media user that uses disinformation to increase tension between different ideas [30].
- **Propaganda**: malinformation that aims to influence the public and a political agenda [31].

However, it should be noted that the field of information disorder is complex and constantly evolving. Other important concepts and terms may not have been covered in this thesis but still contribute to a broader understanding of the problem. For readers interested in

learning more about concepts related to the problem of information disorder, we recommend the following resources [11], [12], [24], [25].

So far, we have provided an in-depth examination of key concepts related to the problem of information disorder, such as disinformation, misinformation, and mal-information, as well as additional concepts relevant to the work presented and crucial to understanding the various aspects of the problem. It is important to differentiate between these terms since the nature and intention behind the spread of false information can vary greatly, which can also affect the effectiveness of countermeasures. In the following chapters of this thesis, we will focus on addressing the problem of misinformation and disinformation and their negative impact on society while respecting the terminology used in the field. However, we will not delve extensively into the complex nuances of the terminology surrounding the issue of information disorder.

2.1.3 The spread of misinformation and disinformation

In this section, we will examine the factors that contribute to the spread of misinformation and disinformation, including the appeal of disinformation, the exposure to disinformation, and the role of social networks in promoting these situations.

The spread of misinformation and disinformation has become a significant concern in recent years, as the rise of social media and the decline of traditional journalism have created a perfect storm for the proliferation of false information [24], [25]. The new era of journalism, characterized by the accessibility of the Internet and the explosion of citizen journalism, where the general public collects, disseminates, and analyses news and information, has undermined fact-checking [21], [32]. Additionally, disinformation has become increasingly appealing as a tool for political manipulation and financial gain. The repeated exposure to false information, facilitated by algorithms on social networks that prioritize engagement over accuracy, has reinforced social biases and created echo chambers [12], [23]. Terminology that we will now explain.

The ease of access to social networks and digital platforms, combined with the power of algorithms to tailor content to individual users, has led to a significant increase in the exposure of people to misinformation and disinformation [11], [12]. Studies have shown that **false information spreads six times faster than true stories** on social media platforms [14], [15]. This is due in part to the fact that humans are more likely to spread false news than the truth. How social media algorithms prioritize engagement and shareability also promotes the spread of false information [33]. Additionally, the appeal of sensational and shocking content leads to users exposed to content that confirms their pre-existing beliefs and biases. This situation eases the creation and distribution of false information and contributes to the proliferation of misinformation and disinformation on social media.

In addition to the appeal of disinformation and the biases present in the spread of false information mention above, the algorithms used by these platforms often lead to a phenomenon known as the "**filter bubble**" where users are exposed to content that aligns with their existing beliefs and interests [12], [24]. This can result in the formation of "**echo chambers**". Echo chambers are online communities where individuals are surrounded by like-minded

individuals who share similar views and beliefs, which contributes to the fragmentation of opinion and polarization [12], [23], [25]. These communities can be based on a specific topic or issue, and within these groups, false information can spread rapidly and be reinforced through the collective belief of the community [14], [15]. This is especially dangerous, as it can make people resistant to counter-arguments or alternative views and entrenched in their beliefs. Studies have shown that false information is more likely to spread in echo chambers and that these communities can amplify misinformation and disinformation [12], [31]. As a result, echo chambers can play a significant role in spreading misleading information and forming false beliefs [14].

Given the current state of affairs, it is clear that this new era of journalism requires new solutions to ounter the spread of false information. One promising area of research is Computational Science, which has proven to be one of the most interested disciplines in the problem of information disorder, alongside Political Science, Sociology and Psychology [12].

2.1.4 Paradigm of fact-checking

In this section of the thesis, we will explore the traditional paradigm of fact-checking as the primary method of dealing with information disorders. We will define fact-checking, including the steps involved and the critical steps of the process. We will also provide examples of organizations that are dedicated to fact-checking and examine how they are organized globally. However, we will also point out the limitations of manual fact-checking after the emergence of social networks, some alternative solutions proposed that highlight the need for computational methods to address this problem.

Fact-checking is the process of evaluating the accuracy and veracity of information, typically in the form of statements or claims made by public figures, organizations, or media resources [12], [23], [25]. This process can be done before and after a claim becomes publicly relevant [24]. This thesis will focus on "ex-post" fact-checking, which refers to evaluating information after it has been disseminated to the public. Ex-post fact-checking is particularly important as it aims to mitigate the impact of misinformation and disinformation on society. In this context, fact-checking is not just to debunk false information but also to provide context and clarification for statements [12], [24].

The fact-checking process typically involves a series of steps [24], [34]. The first step is identifying which major public claims are fact-checkable and should be examined. This is followed by researching and verifying sources to gather the necessary information and best available evidence and, finally, evaluating the credibility of the claims and determining their overall truthfulness.

Fact-checking has grown in relevance and has spread worldwide in the recent decade as the spread of misinformation and disinformation has grown [21], [24]. As a result, many organizations and fact-checking networks have been established to aid in this process. Currently, there are examples of fact-checking organizations around the world, with the United States

being the most representative [24], highlighting Snopes[2], Reuters[3] and PolitiFact[4] among others fact-checkers. Fact-checking is also being done in other parts of the world. Examples of such organizations include Maldita[5] or Newtral[6] (Spain), AFP fact checking[7] (France), Colombiacheck[8] (Colombia), Africa Check[9] (South Africa), BOOM[10] (India).

There are also international organizations like the International Fact-Checking Network (IFCN). These organizations play a crucial role in ensuring the transparency and objectivity of fact-checking. Many of them adhere to a code of principles that guide conscientious fact-checkers in their everyday work. The IFCN, for example, has developed a code of principles that sets standards for accuracy, transparency, and impartiality in fact-checking, and it is widely considered the gold standard for fact-checking organizations [12], [21], [34]. For readers interested in further exploring the different fact-checking organizations associated with the International Fact-Checking Network, we recommend visiting the IFCN's website[11], where a comprehensive list of member organizations can be found.

Manual fact-checking, while essential to ensure the truthfulness of the information, has several limitations [11], [12], [35]. One of the main limitations is the massive volume of information that needs to be fact-checked. Similarly, the time and resources required to investigate each claim are significant constraints. The process is both laborious and intellectually demanding, with estimates suggesting that it takes approximately one day to examine and write a comprehensive report on a factual claim [35]. As stated in the previous section, with the constant influx of information and the speed at which it spreads, fact-checkers struggle to keep up with every claim made. Additionally, manual fact-checking is subject to human error and bias [11], [24], which can lead to inaccurate or incomplete information, inadvertently bringing their own biases to the fact-checking process. Furthermore, manual fact-checking is often reactive, only addressing false information after it has already spread and potentially caused harm [35].

Alternative manual methods, such as crowdsource fact-checking, have been proposed to increase the number of fact-checkers and improve scalability [12]. However, this method is also limited as non-experts may follow different methodologies and are less reliable than experts. Finally, manual fact-checking is not always able to detect advanced disinformation techniques, such as deepfakes and synthetic media, which require specialized tools and techniques to detect [24].

As a result, alternative solutions such as automation are being researched and developed to complement manual fact-checking, address these limitations, and improve its effectiveness, as discussed in Section 2.3.

[2] https://www.snopes.com/
[3] https://www.reutersagency.com/en/
[4] http://www.politifact.com/
[5] https://maldita.es/
[6] https://newtral.es/zona-verificacion/
[7] https://www.afp.com/en
[8] https://colombiacheck.com/
[9] https://africacheck.org/
[10] http://www.boomlive.in/
[11] https://ifcncodeofprinciples.poynter.org/signatories

2.2 Literature Review on Natural Language Processing

As described in the Introduction (Section 1.4), the present thesis employs Transformer models emphasizing semantic and multilingual knowledge to address the issue of false information. It is thus imperative to justify the utilization of this specific architecture throughout the work, as well as the significance of semantics, feature space and multilingualism in the context of the thesis.

In this section, we will explore the field of Natural Language Processing (NLP) and the various techniques used to computationally analyze and comprehend human language. We will begin by providing an overview of the field of NLP and its main goals and applications (Subsection 2.2.1). We will then discuss in subsection 2.2.2 the early NLP techniques used before the advent of deep learning, including rule-based and statistical methods such as n-grams, part-of-speech tagging, and dependency parsing. Then, in Subsection 2.2.4, we will cover Machine Learning (ML) based approaches and statistical methods for vectorization, the required pre-processing step to transform text data into numerical vectors for training. This Section also introduces the use of Deep Learning (DL) based models, including word embeddings, Recurrent Neural Networks (RNNs), and Transformer models. In subsection 2.2.5, we will explore the Transformer architecture, its capabilities and limitations, and its potential as the backbone of the Deep Learning models employed in the thesis. Furthermore, in Chapter 3, we will discuss the importance of semantics, feature space and multilingualism in NLP and why these concepts are incorporated into this thesis. Finally, we will conclude by summarizing the key points covered and the importance of NLP techniques in the context of the thesis.

2.2.1 Introduction to NLP

This subsection provides an overview of the field of NLP and its main goals and applications.

Natural Language Processing (NLP) is a field of artificial intelligence and computer science that is focused on the interaction between computers and human languages to solve practical problems in understanding human languages [36], [37]. The main goal of NLP is to enable computers to understand, interpret, and generate human language, in order to facilitate communication and automate tasks.

The field of NLP has a long history, dating back to the 1950s and 1960s, when the first automatic language translation and conversational systems were developed, such as ELIZA [38]–[40]. However, it was not until the 1990s and 2000s, with the advent of new technologies such as Machine Learning and Deep Learning that NLP truly began to flourish [39], [41], [42]. One of the most notable advancements in this field was the development of neural machine translation, which consists of huge deep neural networks that significantly improved performance compared to the phrase-based statistical approaches that were previously considered state-of-the-art [43].

The field of Natural Language Processing (NLP) continues to grow rapidly owing to advances in human-computer interaction, which now makes NLP techniques easily integrated into diverse fields such as healthcare, finance, and customer service [1], [36], [37], [43]. Some of the most common NLP tasks are Information Retrieval (IR) for automatically extracting

relevant information (e.g., named entities and relations between them) from texts, Machine Translation of text between languages, summarization of documents, automatic answering of questions, categorizing text into predefined categories and clustering of documents [43].

In the context of this thesis, NLP is important because it can be used to automatically identify and flag potential misinformation and disinformation, making the fact-checking process more efficient and accurate [37]. NLP techniques can also be used to analyze the spread and impact of misinformation on social media [14], [15], and to develop computational solutions for addressing the problem of misinformation.

2.2.2 Early NLP techniques

This subsection describes the early NLP techniques used before the emergence of machine learning and deep learning.

Early NLP techniques were mainly based on statistical and rule-based methods [44], [45]. These rule-based methods were used to analyze and understand texts by manually creating a set of rules determined by the linguistic structure of the language, such as grammar and syntax [46]. However, this method had its limitations, as the rules needed to be manually created for each language and were often complex and challenging to maintain. For instance, it was difficult to decide the best way to parse and interpret ambiguous sentences, as the rules were often too rigid and did not account for many exceptions [47].

Part-of-speech tagging and dependency parsing are also standard techniques used in early NLP [44], [46]. Part-of-speech tagging consists of identifying the grammatical role of each word in a sentence, such as a noun, verb, adjective or pronoun. On the other hand, dependency parsing involves analyzing the grammatical relationships between words in a sentence [48]. Part-of-speech tagging and dependency parsing are also standard techniques used in early NLP [44], [46]. Part-of-speech tagging consists of identifying the grammatical role of each word in a sentence, such as a noun, verb, adjective or pronoun. On the other hand, dependency parsing involves analyzing the grammatical relationships between words in a sentence [48]. Both of these techniques can be applied to tasks such as text summarization and machine translation, as they help understand sentence structure.

Another technique commonly used in early NLP was statistical methods, which involved developing probabilistic models to analyze text [46], [48]. One popular method used in statistical NLP is n-grams, which involves breaking down the text into sequences of words and analyzing their frequency and distribution [45]. For example, a bigram (n=2) would be a sequence of two words. This method were applied to various NLP tasks such as language modelling, text classification, and keyword extraction for information retrieval.

The statistical models common in the early days of NLP were Gaussian Mixture Models (GMM), Hidden Markov Models (HMM) and Conditional Random Field (CRF) models [49], [50].

Probabilistic Gaussian mixture models (GMMs) were used in speech recognition by exploiting their ability to cluster features by modelling signal variations, assuming that data points are generated from a mixture of Gaussian distributions. Hidden Markov Models (HMMs)

are also generative models. However, they consider the succession of features and predict the most likely sequence of labels by modelling the joint probability of sequences, such as words in a sentence and phonemes in speech. Finally, Conditional Random Fields (CRFs) are discriminative models that also consider the entire sequence better suited for classification tasks, modelling the conditional probability and decision boundary between classes.

Although the initial statistical NLP methods were considered state-of-the-art, they had limitations. These methods struggled with complex linguistic tasks and could be computationally expensive when working with large datasets. Moreover, some of these methods assumed observation independence, which is not the case in text observations (words) since they are often not independent. These methods also heavily relied on the quality of the features, which required good manual design of preprocessing methods to obtain valuable features that change in every scenario (see Figure 2.3). However, these early methods laid the foundation for more advanced techniques used in NLP today [43], [51].

2.2.3 Tokenization and Vectorization Transformation (Embeddings)

Before we move on to more complex NLP models and explore how the NLP has evolved, it is crucial to focus on an often overlooked aspect: tokenization. Understanding its essential role, the difference from vectorization, and the relevance of coupling them during training and inference will allow us to gain a better understanding of the advancements in NLP and the significant impact of Transformers.

To understand these concepts, the Figure 2.2 will be helpful, which presents an overview of a NLP workflow. Although this figure illustrates the workflow using the BERT model as an example, the fundamental steps of tokenization, vectorization, and embedding generation are common to many NLP approaches, although the specific implementations may vary.

Tokenization

As can be seen, the primary purpose of tokenization is to transform unstructured text data into a structured numerical form (step 1 up to 4). This is essential because computers require data in a structured, numeric format to perform any operation.

Tokenization involves dividing text into smaller units, such as words, characters, or sub-words, known as tokens (step 1 to 2). A visual representation of this process and examples of different types of tokenization are shown in Table 2.1. Once the text is broken down into tokens, each token is then mapped to a unique ID based on a predefined vocabulary, transforming the text into a sequence of numeric values (step 3). This way, the tokenization process has converted raw text into input features (step 4) that an AI model can understand. Subsequently, these numeric values are converted into dense embeddings through the vectorization process, which capture semantic information, allowing the model to understand and process the text in its complex latent space for various tasks (step 5).

Before moving to the following subsection about this vectorization process, we will present various tokenization methods [52]. We will discuss tokenization approaches in more detail in Section 2.2.6 as we explore the multilingual capabilities of Transformer models.

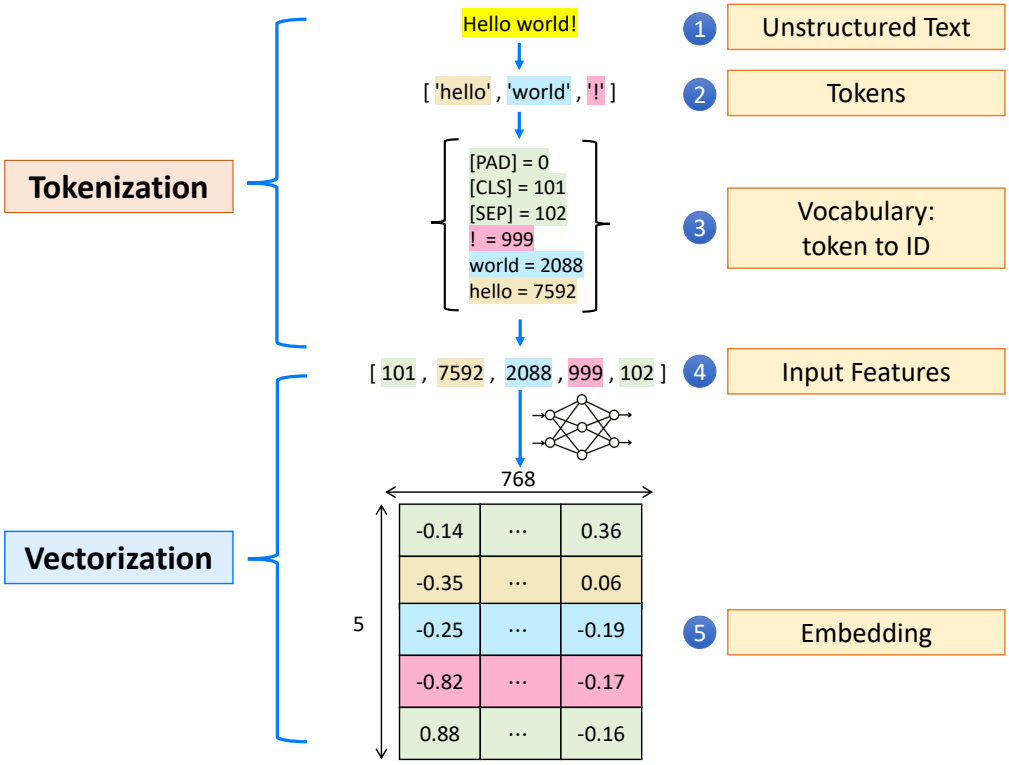


Figure 2.2: This diagram illustrates the workflow of NLP using BERT as an example model. The general steps of NLP involve tokenization, where text is segmented into tokens; vectorization, which converts tokens into numerical IDs; and embedding, where numerical IDs are transformed into dense vectors. [CLS], [SEP] and [PAD] are special tokens representing the beginning, the end and padding elements of a sequence.

- **Word-Level Tokenization**: This method divides the text into individual words. It is effective for languages with clear word boundaries but may miss nuances in languages with compound words or complex structures.
- **Morphological-Level Tokenization**: This method segments text into morphemes, the smallest grammatical units in a language. It is especially valuable for languages with words that can take on multiple forms based on tense, case, and other factors.
- **Character-Level Tokenization**: This method breaks down the text into individual characters. It is particularly useful for tasks that require a deep understanding of the text structure, such as certain types of text generation or languages with no clear word boundaries.
- **Sub-word level Tokenization**: represents a balance between word-level and character-level tokenization. It efficiently handles rare words and is commonly used in modern NLP Transformer models due to its adaptive handling of a wide range of vocabularies and linguistic structures. One example of this type of tokenization algorithms is Byte Pair Encoding (BPE) [53].

Method	Tokenized Text
Word-level	['This', 'Thesis', 'is', 'about', 'Computer', 'Science', 'at', 'the', 'UPM']
Morphological-level	['thi', 'thesi', 'is', 'about', 'comput', 'scienc', 'at', 'the', 'upm']
Character-level	['T', 'h', 'i', 's', ' ', 'T', 'h', 'e', 's', 'i', 's', ' ', 'i', 's', ' ', 'a', 'b', 'o', 'u', 't', ' ', 'C', 'o', 'm', 'p', 'u', 't', 'e', 'r', ' ', 'S', 'c', 'i', 'e', 'n', 'c', 'e', ' ', 'a', 't', ' ', 't', 'h', 'e', ' ', 'U', 'P', 'M']
BPE	['This', 'ĠThe', 'sis', 'Ġis', 'Ġabout', 'ĠComputer', 'ĠScience', 'Ġat', 'Ġthe', 'ĠU', 'PM']
WordPiece	['this', 'thesis', 'is', 'about', 'computer', 'science', 'at', 'the', 'up', '##m']

Table 2.1: Example of different tokenization methods for the sentence "This Thesis is about Computer Science at UPM".

Vectorization Transformation (Embeddings)

As explained earlier, after dividing the text input into individual tokens and assigning them a unique ID, the unstructured text is transformed into a vector representing the initial features which are used as inputs for an AI model. The next step is vectorization, which involves creating and converting the vector or embedding in order to extract and condense information for various purposes.

As we will cover, initially, vectorization involves basic methods like counting tokens or analyzing statistical properties suitable for traditional models. These methods have evolved into more sophisticated techniques, where tokens are indexed by their position in a vocabulary and embedded in a feature space that encapsulates linguistic knowledge more deeply.

This evolution, which began with the complex and labour-intensive early stages of rule-based, context-independent representations, has significantly enhanced the models' capacity to discern linguistic subtleties without the need for explicit rule articulation [46], [49]. The initial stages required manual intervention for feature extraction, constrained by the limitations of pre-established rules (see left panel from Figure 2.3). This approach was gradually replaced by a phase where features extracted through statistical analysis enabled pattern recognition by machine learning models, albeit remaining somewhat detached from a profound linguistic comprehension (middle panel from Figure 2.3).

The latest advancements (see right panel from Figure 2.3), which represent a significant breakthrough in the field, incorporate these representations directly, enabling models to manipulate and interpret text in a way that closely mirrors human cognition, recognizing context and semantic relationships [54]. This progression underscores the shift towards models that dynamically extract linguistic knowledge, increasingly incorporating context in text interpretation.

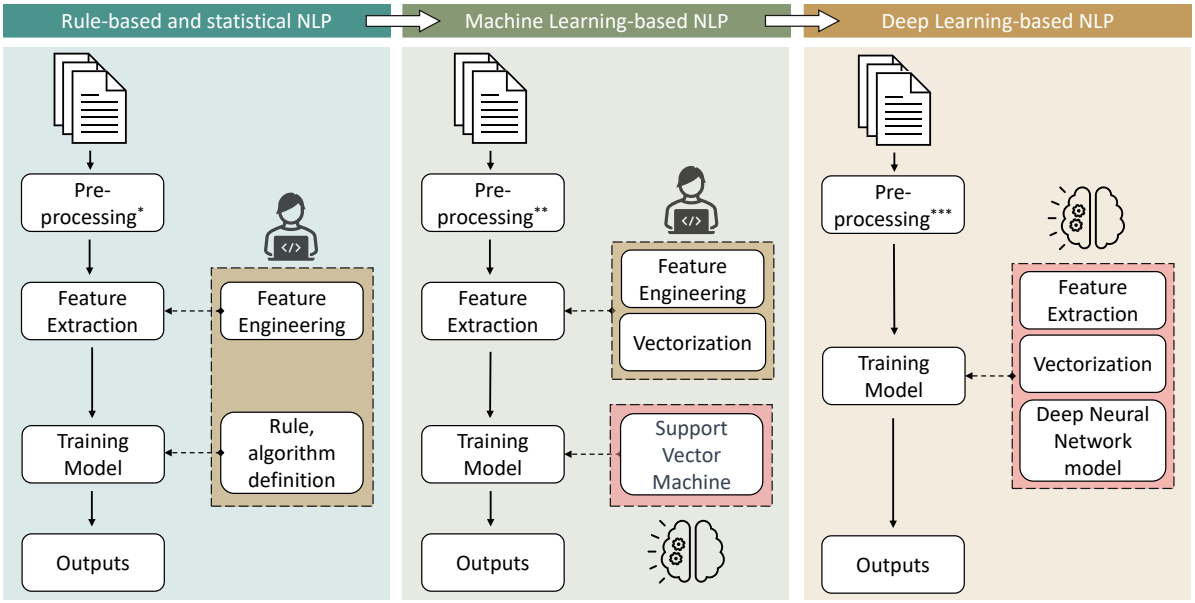


Figure 2.3: Evolution of Natural Language Processing - The workflow of early NLP techniques (left) relies on pre-defined rules and manual annotation, while Machine Learning-based approaches (center) require pre-processing of text (e.g., stopwords removal) into numerical vectors before training a model (e.g., SVM). In contrast, Deep Learning-based NLP approaches (right) involve the use of deep neural networks for both vector representation and training, allowing for the automatic learning of features from data and improved handling of context and semantics. (Adapted from: [54])

2.2.4 Machine Learning and Deep learning Embeddings in NLP

Building on the concepts of tokenization and vectorization, this section explores the evolution of NLP and the role of context in text interpretation.

As depicted in Figure 2.4, we will start by exploring **traditional context-independent** techniques such as Bag of Words and TF-IDF, which are fed into Machine Learning models to extract patterns from static representations that do not consider the context.

Next, we will discuss **advanced context-independent** embeddings, such as Word2Vec. Despite being context-independent, they enhance context integration through shallow models or co-occurrence matrices. These models create global static embeddings that capture the meaning of a word in different contexts and the surrounding words. However, these embeddings remain static after being trained.

Finally, we will discuss **context-dependent** methods, which, unlike their predecessors, generate unique embeddings for words based on their interaction with the other text elements, changing based on the surrounding words in a sentence even after being trained, this means that the same word will practiaclly have a different embedding for each possible sentence where it appears. This approach represents a significant advance towards models that extract linguistic knowledge dynamically, progressively integrating context into text interpretation.

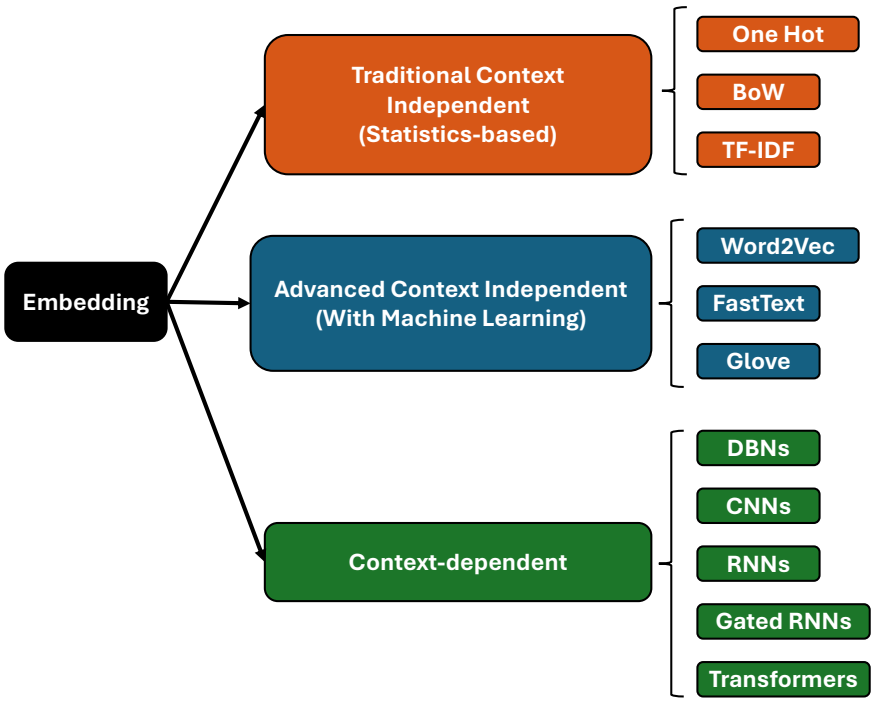


Figure 2.4: Taxonomy of types of embeddings according to the role of context in text interpretation (Adapted from [51]).

Traditional Context-Independent Static Embeddings: One-hot, Bag of Words and TF-IDF

The terms "context-independent" and "static" refer to techniques where each token is directly translated into a vector based on frequency or TF-IDF scores and then fed to a Machine Learning model to extract the partners. In these techniques, word order and context are ignored - only word frequencies matter, and words are considered independently.

One Hot Encoding

One of the most basic techniques used to represent data numerically is the one-hot encoding technique [55]. In this method, a vector is created with a length equal to the number of unique words in the data, where an index in the vector represents each word. For each word, the corresponding index is assigned a value of 1, while all other indices are assigned a value of 0 representing absence. However, while this approach is simple, it does not capture any semantic or statistical relationships between the words in the data and suffers from high sparse dimensionality.

Bag of Words (BoW)

The Bag of Words (BoW) model, called Count Vectorization, generates vectors representing texts by counting how frequently each word occurs within a text [56]. This technique, shown in Figure 2.5, goes beyond merely indicating the presence or absence of words, as done by one-hot encoding, and instead quantifies the frequency of each word's appearance. However, BoW still has its limitations as it treats all words equally, not considering multiple meanings for the same word (i.e. polysemy), and is susceptible to counting overrepresented common words such as "the" and "and", which can lead to inaccurate results when working with sentences with similar meanings but different words.

Term Frequency-Inverse Document Frecuency (TF-IDF)

To overcome this, the alternative tecnique TF-IDF [57], [58] defines importance of a term by taking into consideration the importance of that term in a single document, and scaling it by its importance across all documents. The TF-IDF equations are the following:

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \tag{2.1}$$

$$\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents}}{1 + \text{Number of documents containing term } t}\right) \tag{2.2}$$

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \tag{2.3}$$

Where:

- t represents a term within document d,
- d is a specific document within the corpus D,

A- the banks of the river are flooded
B- the bank has bank loans.
C- the dogs are cute
D- the bears are scary
E- the bank transactions are digital

One hot

Sentence	are	bank	banks	bears	cute	digital	dogs	flooded	has	loans	of	river	scary	the	transactions
A	1	0	1	0	0	0	0	1	0	0	1	1	0	1	0
B	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0
C	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0
D	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0
E	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1

Bag of Word

Sentence	are	bank	banks	bears	cute	digital	dogs	flooded	has	loans	of	river	scary	the	transactions
A	1	0	1	0	0	0	0	1	0	0	1	1	0	**2**	0
B	0	**2**	0	0	0	0	0	0	1	1	0	0	0	1	0
C	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0
D	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0
E	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1

TF-IDF

Sentence	are	bank	banks	bears	cute	digital	dogs	flooded	has	loans	of	river	scary	the	transactions
A	0.25	0.00	0.44	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.44	0.44	0.00	0.42	0.00
B	0.00	0.73	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.45	0.00	0.00	0.00	0.22	0.00
C	0.35	0.00	0.00	0.00	0.63	0.00	0.63	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.00
D	0.35	0.00	0.00	0.63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	0.30	0.00
E	0.32	0.45	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.27	0.56

Figure 2.5: Comparison of One-Hot, Bag of Words and TF-IDF textual feature extraction techniques. The highest values per sentence (row) are highlighted to illustrate how the feature extraction techniques differ.

- D signifies the entire collection of documents.

Term frequency $\mathrm{TF}(t, d)$ (equation 2.1) calculates the frequency of term t in document d, normalized by the total number of terms in d.

Inverse document frequency $\mathrm{IDF}(t, D)$ (equation 2.2) assesses the importance of t across the entire corpus D, offset by adding 1 to avoid division by zero and logarithmically scaled to ensure that terms common across many documents are penalized.

$\text{TF-IDF}(t, d, D)$ (equation 2.3) combines the TF and IDF scores to evaluate the importance of term t in document d relative to the corpus D balancing its frequency in a specific document against its commonness across the entire corpus. This way, a high weight in TF-IDF is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. For example, we can check in the Figure 2.5 that words like 'the' that naturally appear in the English language are discarded by giving it a lower IDF value compared to the rest of the preivous techniques.

As a picture is worth a thousand words, Figure 2.5 shows precisely the difference between these models across various sentences, highlighting the words assigned the highest weight in each sentence. Compared to One-Hot Encoding, it can be observed how Bag Of Words incorporates more information in the encoding of the sentences according to the frequency of unique words in each sentence, highlighting, for example, the word "bank", a detail missed by One Hot

Encoding. Nonetheless, both techniques tend to accentuate words of little semantic value, such as "the". This issue is resolved by TF-IDF, where we observe that words highlighted are more informative and specific to each sentence, with greater granularity, effectively distinguishing "bank" while diminishing the significance of common words like "the".

However, a significant limitation remains: "bank" and "banks" are treated as distinct entities due to their purely frequentist representation. To solve this problem, methods such as lemmatization or stemming arise [59], simplifying words to their root form, thereby, for instance, converting plurals to singulars to unify their representations, but it is still a frequentist view.

Machine Learning

These feature extraction techniques generate the initial input features fed into the tabular machine learning models to perform an NLP task. As shown in Figure 2.3, this is an advance over early statistical methods (Section 2.2.2), as machine learning models learn complex patterns from textual data features, rather than relying on predefined rules [45], [60]. This was a breakthrough as well-known models, such as Naive Bayes or Support Vector Machines, can process large amounts of data to improve textual understanding. Despite this, this strategy has the limitation of **separating textual feature extraction from the model learning process** [54].

To recap, major shortcomings of the ML traditional approach includes (1) initial vectors from the feature extraction often lack the ability to capture semantic meaning and limits the potential of what the ML model can learn [43], (2) an inability to account for polysemy, where the same word may have multiple meanings depending on context, and (3) no link between the initial feature extraction process (tokenization) and the vectorization process within the ML models.

In the next section, we will see how this limitation can be overcome to generate more semantic meaningful textual features representation, by capturing context, the interaction between words, and the integration of tokenization and vectorization processes.

Advanced Context Independent Global Static Embeddings: Word2Vec and FastText

The next advance was context-independent global word embedding. Again, these are static embeddings, meaning they remain the same regardless of the context of the text. However, the innovation lies in incorporating more semantic knowledge in this static representation through a "global" view of all the situations in which a term can be found. This attempt helps to address polysemy by considering all possible situations and incorporating them into a singular static embedding [62].

Word2Vec

In 2013, Mikolov et al. [61] proposed a new method for learning distributed representations of words, known as “word embedding” or “word vectors”. The technique, Word2Vec, uses a shallow neural network with an encoder-decoder structure pre-trained on unlabeled text corpora. The unsupervised training process involves reconstructing a target word based on

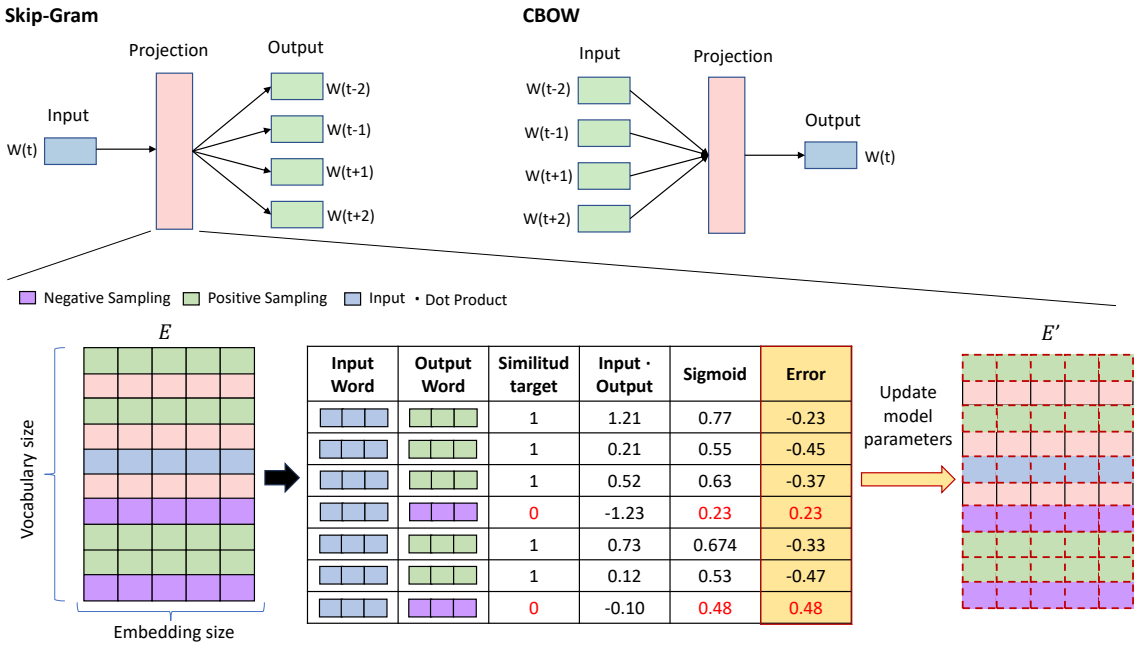


Figure 2.6: Diagram illustrating the Word2Vec CBOW and Skip-Gram architectures: showcasing the process of embedding optimization through positive and negative sample comparison, guided by logistic loss to update word representations (E to E') (Adapted from: Mikolov et al. [61] and Jay Alammar from jalammar.github.io, 2019).

its context or predicting context words given the target word, as shown in Figure 2.6. This approach allows for the creation of vectors for each word that capture both syntactic and semantic relationships, resulting in similar representations for words with similar meanings [43].

In a more detailed overview, the setup of Word2Vec starts with creating a unique vocabulary from the training corpus with randomly initialized word embeddings, adding an "Out Of Vocabulary" (OOV) embedding allocated for those not included in the vocabulary. This vocabulary is created with a simple tokenization based on split text into words. Then, the training updates the word embeddings as they recur in varying contexts, thereby embedding global linguistic knowledge into each word's vector, which remains static post-training [62]. For this training purpose, Word2Vec can use two neural architecture approaches as previously mentioned: Continuous Bag of Words (CBOW) and Skip-Gram [62]. In CBOW, the model predicts a target word based on its context words. Skip-Gram, conversely, predicts context words given a target word.

Take the Skip-Gram architecture as an example; it consists of a shallow network structure with typically just one hidden layer between the input and output layers. As illustrated in Figure 2.6, this hidden layer corresponds to the Embedding matrix (E), with dimensions proportional to the vocabulary size and the embedding size. During a training iteration, the model selects an input word from the text using its associated vector from the E matrix. Concurrently, it retrieves the embeddings for the context words (positive samples) and randomly chosen words not in the input word's context (negative samples), all from the same E matrix.

The training involves calculating the dot product followed by a sigmoid operation between the

input word vector and both sets of output word vectors. For true context pairs, the model aims for the sigmoid dot product to approximate 1, signifying a high similarity, whereas, for negative samples, it targets a value near 0, indicating low similarity. The discrepancy between these expected outcomes and the model's predictions, quantified by a logistic loss function, guides the optimization of the E matrix (neural network hidden layer parameters), effectively refining the word embeddings for the words that incorporate a more global view of the word's use across various contexts.

FastText

The success of Word2Vec led to the use of more complex neural network architectures and its extension for representing sentences and documents as embeddings [63], [64]. Subsequently, several improved solutions have been proposed, such as Global-Vector (GloVe) [65] which utilizes statistical information computed on the entire corpus, and FastText [66], [67] which injects sub-word (character n-grams) information to describe the internal structure of a word.

FastText was developed by Facebook and it extends Word2Vec by considering character n-grams in addition to words. Unlike Word2Vec, which operates at the word level. A character n-gram is a set of co-occurring characters within a given window.

Tokenization at character n-grams allowed FastText to overcome Word2Vec's limitations: its dependency on a predefined vocabulary. In Word2Vec, words not encountered during training are mapped to the OOV token previously mentioned, potentially losing valuable information. FastText addresses this by representing each word as a sum of its constituent n-grams. This technique fosters parameter sharing among words with similar morphological features, enriching the model's ability to understand and represent them. So also, when certain words are missing from the training vocabulary or rarely occur, we can still have a representation for them if their n-grams are present as part of other words.

Advancements and Remaining Gaps

These context-independent global embedding techniques, specifically through Word2Vec and FastText, brought remarkable advancements to NLP, addressing several limitations of previous techniques. A significant advancement was integrating tokenization and vectorization directly into the models, enabling effective feature extraction and contributing directly to the learning capabilities of the models. Moreover, FastText's sub-word level tokenization approach significantly advances the models' ability to generalize to new or unusual sentence structures and manage words not seen during training.

However, despite these advancements, limitations remain. Both Word2Vec and FastText struggle with fully capturing polysemy, the phenomenon where a word may have multiple meanings depending on the context, due to the static nature of their word embeddings [45], [63]. Additionally, the fixed embeddings do not consider the grammatical structure of sentences or the dynamic contexts in which words appear, which can deteriorate understanding of complex sentence meanings [43]. Finally, while these models offer improved semantic extraction capabilities, they primarily focus on individual words rather than the sequential nature of words that compose entire sentences.

Context Dependent Dynamic Embeddings: DBN, CNNs, RNNs, LSTMs

Context-dependent methods, in contrast to the previous traditional and advanced context-independent word embeddings, encode distinct embeddings for the same kind of words based on the contextual information where it is used [51].

Context-dependent embeddings, while also originating from a unique global vocabulary and integrating tokenization and vectorization processes within the model framework, diverge significantly from the previous methods by addressing the limitations related to word context. As mentioned above, context-independent methods generate a single, static representation for each word, failing to capture the details of language use across different contexts. Context-dependent methods, however, learn sequence-level semantics, dynamically analyzing the sequence and interactions of words within texts, allowing for the generation of distinct representations of the same word when used in different contexts. This approach effectively resolves the challenge of polysemy by providing versatile, context-aware embeddings for words based on their usage.

In this context-dependent embedding, Deep Learning models are the predominant models. Among them, we will focus on different DL architectures, such as Deep Belief Networks (DBNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long-Term Short-Term Memory (LSTMs).

Deep Belief Networks - Unsupervised fully connected generative models

Deep belief networks (DBNs) proposed by Hinton et al. in 2006 [68] belong to the generative models family that leverage the power of simpler networks such as Restricted Boltzmann Machines (RBMs) [69] and autoencoders by stacking them hierarchically.

As well-explained in [51], the critical component of DBNs is the RBMs. RBMs are a type of generative stochastic neural network composed of two layers of neurons: a visible layer that interacts with the input data and a hidden layer that captures latent features. Each neuron in these layers is fully connected to all neurons in the preceding layer (i.e., the connection between visual and hidden neurons), while there are no connections between neurons within the same layer (i.e., the interconnection between hidden neurons). The main objective of RBMs are to model input data distribution on the visible layer by training it to reconstruct the given input while learning a probabilistic representation in the hidden layer.

The DBN backbone architecture involves sequentially linking RBMs, where the hidden layer outputs of one RBM serve as the visible layer inputs of the next, a visual representation is depicted in Figure 2.7. This modular approach allows DBNs to model increasingly abstract data representations at each layer, starting from basic data structure at the initial layer and advancing to complex, higher-level features in subsequent layers. As an input progresses through DBN stacked layers, they transition from capturing fundamental data attributes to identifying intricate patterns. For readers interested in visualizing DBN architectures and more related concepts, we recommend the following resource chapters [70], [71].

DBNs are layer-wise unsupervised models as each RBM within the network is trained independently through contrastive divergence (CD) [72]. The CD process iteratively updates the network's weights by alternating between forward passes that map visible units to hidden

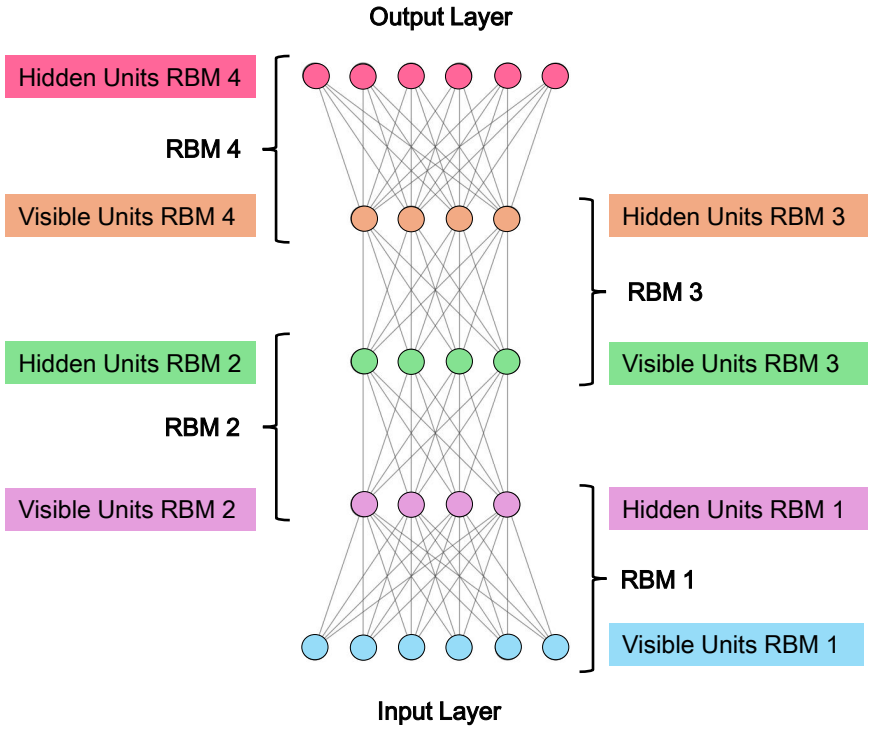


Figure 2.7: Schematic representation of Deep Belief Neural Network (Adapted from [51]).

representations and backward passes or 'reconstruction steps' that aim to regenerate the visible units from the hidden layer activations. This iterative refinement approximates the log-likelihood gradient of observing the training data, optimizing the model's parameters for accurate data generation and feature extraction [70].

Recapitulating, DBNs successfully compute context-dependent representation by employing an approach based on a) layer-wise pre-training where each layer learns to capture increasingly abstract representations (b) use of unsupervised learning algorithms to maintain information from input and finally (c) the fine-tuning the entire network through backpropagation. This structured approach enables DBNs to learn complex, hierarchical features of data, mirroring the unsupervised learning nature seen in models like Word2Vec and FastText, but with an enhanced ability to model sequence-level semantics and dynamically represent polysemous words based on context.

However, despite these strengths, DBNs face certain limitations [71]. While effective in optimizing network weights, their training process can be computationally demanding, particularly as the network size and depth increase. This computational intensity is due to the fully connected layers between the visible and hidden units. Moreover, while DBNs are predominantly unsupervised, applying them to supervised tasks such as classification requires additional steps, like training a classification layer on top of the pre-trained, unsupervised model, being more related to generative NLP application tasks.

Convolutional Neural Networks - Localized Pattern Recognition

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were the two

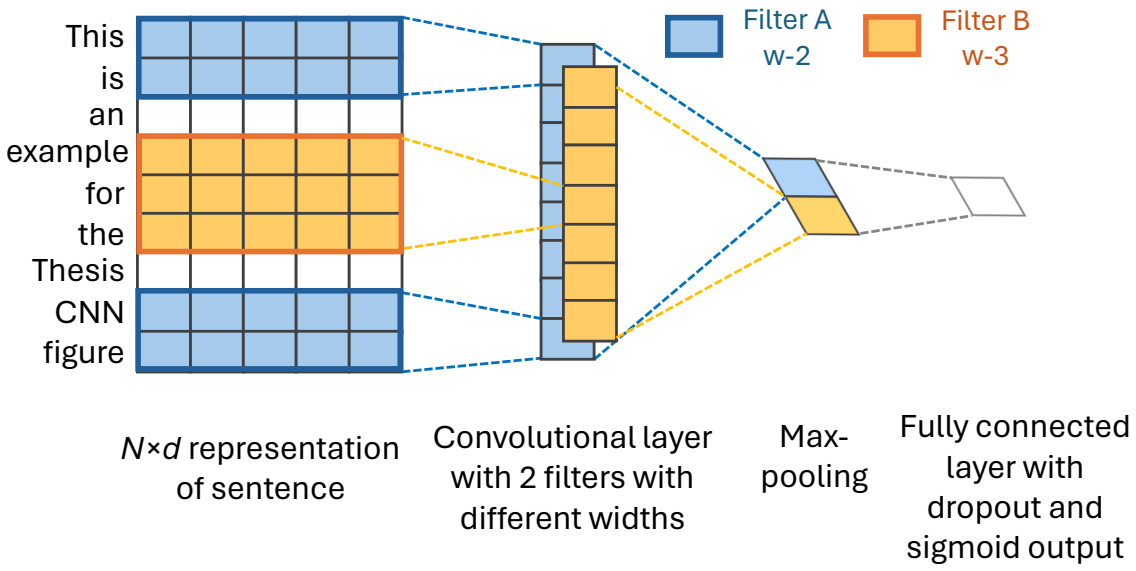


Figure 2.8: Visual representation of the insights of a CNN applied in NLP for binary text classification with different filters. N represents the sequence length, and d the input dimension before convolution (Adapted from [76])

primary deep learning architectures before the emergence of Transformers in 2017. CNNs are hierarchical structures similar to Deep Belief Networks (DBNs), and they excel at recognizing patterns in images due to their spatial hierarchy. This spatial hierarchy also makes CNNs useful for natural language processing (NLP) applications.

Unlike DBNs, CNNs share weights across the network, significantly reducing the number of parameters compared to a fully connected network like a DBN. This makes them more computationally efficient, especially for large images [73].

In NLP, CNNs adapt to capture the spatial relationship between words and identify meaningful patterns within the text, treating sequences of words or characters much like the pixels in an image. The use of CNNs in NLP is illustrated in Figure 2.8 and extensively explained in [74], [75].

Convolutional Neural Networks (CNNs) for text processing begin with an input layer where sequences are represented as dense vector feature maps (n words by k dimensions) , encapsulating the dimensionality of both words and sequence length. The core of CNNs lies in the convolution layer, where multiple filters slide across the input feature map to detect patterns and features within fixed-size text segments, effectively learning from sliding window w-grams. This is followed by a pooling layer, typically employing max pooling to distil these detected features into a more compact and significant representation, achieving spatial invariance by focusing on the most salient features irrespective of their original position in the text. Finally, fully connected layers integrate these features for task-specific outputs, and dropout layers are incorporated to prevent overfitting by randomly omitting subsets of features during training [75], [77].

This process allows CNNs to efficiently capture and analyze local contextual patterns within

the text, making them adept at tasks requiring understanding complex linguistic structures. Hence, CNNs are proficient at extracting local context and identifying intricate patterns within closely arranged words, and CNNs can model long context dependencies [74].

Despite these advantages, CNNs face challenges in sequential data analysis [51]. Their inherent design, focusing on immediate neighbours within a fixed scope, may overlook longer-range dependencies critical for a comprehensive understanding of the text. This limitation underscores the need for models that can seamlessly navigate through extended sequences for a fuller contextual grasp. So, even though they suppose a huge advance remarking long dependencies, their capability for extracting semantic and modelling natural language nuances still needs to be fully achieved solely by CNNs. They must rely on other architectures that successfully manage the sequential nature of language, RNNs.

RNNs - Sequential Data Processing with Memory

Compared to CNNs' hierarchical pattern recognition, RNNs excel in semantically understanding sequences by processing data sequentially, such as text or time series, where context and order are crucial. The architecture of RNNs is precisely designed to maintain a "memory" of past inputs via hidden states, which is crucial for tasks like language modelling, where understanding each word can depend on its predecessors far back in the sentence [74], [75].

As shown in Figure 2.9, the core of RNNs lies in their structure, which, unlike CNNs, loop back on themselves, a property often described in the literature through the terms 'rolling' and 'unrolling' the network [64]. This looping mechanism enables RNNs to carry forward an internal state or memory from one input to the next in a sequence. At each timestep, RNNs process input tokens (x_i) (e.g., words in a sentence), updating their hidden state (h_i) by integrating both the new input and the contextual information accumulated from previous inputs. This continuous update mechanism allows RNNs to capture temporal relationships and context effectively, enabling a more detailed understanding and generation of sequences. An enhanced version of RNNs architecture are the Bidirectional RNNs (Bi-RNNs) which each timestep's hidden state comprises information from both the past and the future relative to the current position in the sequence, enabling the network to capture context from both past and future inputs. This dual-direction processing is particularly advantageous for tasks where the understanding of each sequence element can benefit from the context provided by the entire sequence, not just the elements preceding it, like text classification.

One advantage of RNNs is their capacity to process variable-length sequences. This flexibility is particularly crucial for language modelling, where sentences can vary significantly in length, and each word's meaning may depend on its immediate context and words that appeared much earlier in the sequence. Traditional architectures like CNNs, with their fixed input sizes and local receptive fields, struggle to capture such dependencies, especially over long distances. Both CNNs and RNNS have advantages, where CNNs are hierarchical and RNNs sequential architectures. Which architecture performs better depends on how important it is to semantically understand the whole sequence, being RNNs the one that excel on this [74].

Nevertheless, RNNs' sequential processing, characterized by the accumulation of contextual information through time, is both a strength and a challenge. While Backpropagation Through Time (BPTT) [78] allows for the dynamic adjustment of model parameters based on past

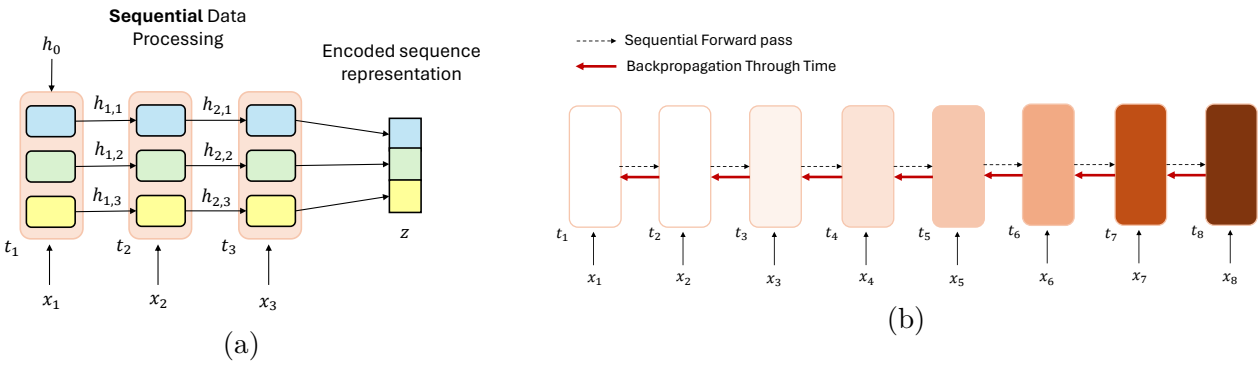


Figure 2.9: (a) An example of an unrolled many-to-one RNN with three units. Each unit processes an input (x) at its respective time step (t), updating its unique hidden state (h). Despite shared parameters across units, each maintains distinct hidden states throughout the sequence. The final output (z) represents the encoded representation of the entire input sequence, synthesizing the information accumulated across all time steps. (b) Vanishing gradient visualization where the contribution from the earlier steps becomes insignificant in the backpropagation gradient. Colour intensity represent the magnitud of the gradient

errors during training, it also introduces the complexities of vanishing and exploding gradients in long texts, where the contribution of information decays exponentially or explodes over time as the gradient is calculated through a long chain of recurrence [51]. Figure 2.9 illustrate this phenomenon of vanishing gradient in the unrolled illustration of an RNN unit, where the contribution from earlier steps becomes insignificant in the gradient while computing backpropagation.

Gated RNNs: LSTMs and GRUs

To address these limitations, advancements in RNN architecture led to the development of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) cells. Both architectures incorporate gating mechanisms to regulate the flow of information, allowing the network to retain important information over long sequences and discard irrelevant data. For the sake of space, we have not included a visual representation of these architectures, but for any reader interested, this illustrated explanation from Andreas Madsen is highly recommended [79].

LSTMs, introduced by Hochreiter and Schmidhuber in 1997 [80], are a sophisticated extension of the traditional RNN architecture designed to enhance memory retention. They achieve this through a complex arrangement of gates: the forget gate decides which information to discard from the cell state; the input gate determines which new information is stored in the cell state; and the output gate decides what information to pass to the output. This structure allows LSTMs to mitigate the vanishing gradient problem effectively.

On the other hand, GRUs, proposed by Cho et al. in 2014 [81] are an alternative to LSTMs, which combine the forget and input gates into a single update gate and merge the cell state and hidden state, simplifying the network architecture and reducing the computational requirements. Despite their simplified structure, GRUs have shown performance comparable

to LSTMs in many tasks.

While LSTMs and GRUs address the limitations of basic RNNs in capturing long-term dependencies, they introduce their own challenges, notably increased computational complexity and parameter intensity [43], [51]. Research has shown that this approach of sending the last hidden state turns out to be a bottleneck for long sentences, and the intricate gating mechanisms of LSTMs and GRUs require a significant amount of computational resources, which involve nonlinear functions such as sigmoid and tanh to regulate the flow of information. These functions, while crucial for determining what information to retain, discard, or pass along, significantly augment the computational demands during both the forward and backward passes through the network. Consequently, models that utilize LSTMs or GRUs for dealing with large sequences or datasets require longer training times and the necessity for more powerful hardware.

2.2.5 Advancements in NLP: Attention Models and Transformer architectures

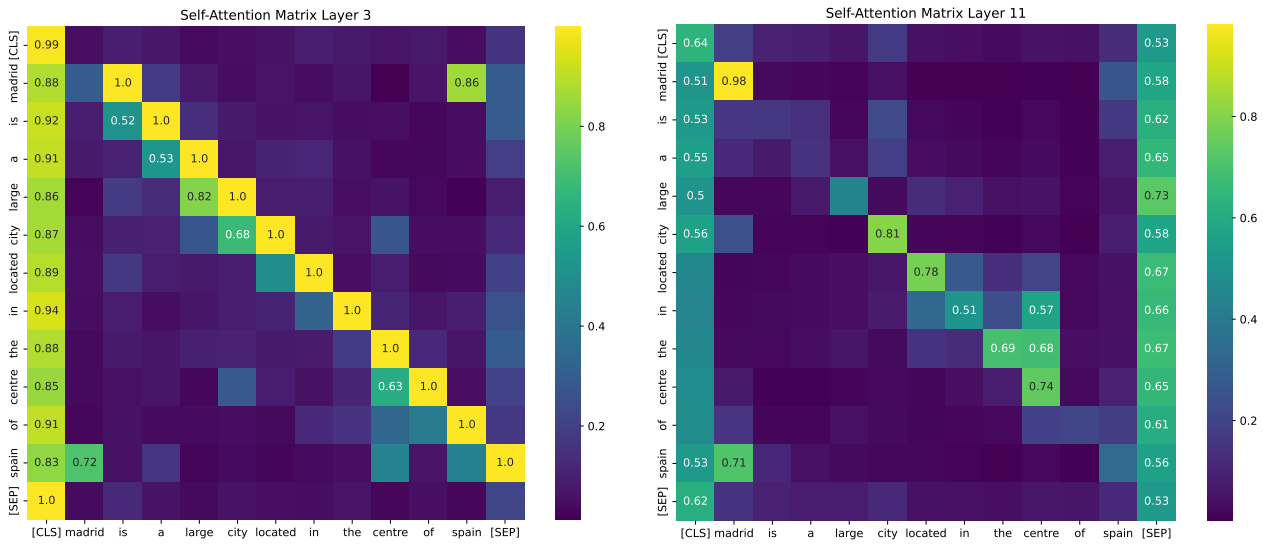
Introduction

So far, we have showcased the remarkable evolution of NLP, progressing from statistical and rule-based models to traditional context-independent static embeddings for feature extraction in Machine Learning models (see 2.2.4). Subsequently, shallow models like word2vec introduced more context into token representations by providing context-independent global representations (see 2.2.4). This paved the way for deep learning architectures that encode context-dependent embeddings, capable of generating distinct representations for the same tokens based on their surrounding context (see 2.2.4). The deep learning models presented include Deep Belief Networks (DBNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Gated RNNs, where each architecture has contributed to improving semantic and linguistic modelling, offering unique advantages and drawbacks, ranging from hierarchical to spatial to sequential architectures. Nevertheless, there remained room for further improvement in capturing long-range dependencies within sequences and efficiently parallelizing the process.

The limitations of traditional sequence models led to the emergence of the Attention mechanism, a technique that allows models to selectively focus on relevant parts of the input sequence when computing a representation for a specific output. This mechanism proved to be a game-changer, as it enabled models to better understand context and capture long-range dependencies, surpassing the capabilities of previous architectures.

Building upon the success of the Attention mechanism, the Transformer model was introduced, which revolutionized the NLP landscape. There is no doubt that Transformers and Attention mechanisms have established the state of the art with excellent ability to understand context, model semantics and linguistics, and enable parallelization.

In this subsection, we will first explore the Attention Mechanism 2.2.5, covering its core concept, mathematical formulation, and parallelization capabilities. Next, we will explain the Transformer Architecture 2.2.5, providing an overview, and examining the self-attention



(a) (b)

Figure 2.10: Visualization of self-attention matrices from different layers of a Transformer BERT-base model fine-tuned for named entity recognition on the sentence *"Madrid is a large city located in the centre of Spain"*. [CLS] and [SEP] are tokens added to the start and end of the sequence as a prefix and sufix. (a) The self-attention matrix in Layer 3 shows the model capturing long-range dependencies like the relationship between *"Madrid"* and *"Spain"* (0.72 and 0.86 values) . (b) In Layer 11, the multi-head attention mechanism allows the model to additionally focus on other contextual relationships like "center", *"city"* and *"located"*, while still maintaining the *"Madrid"-"Spain"* dependency. The heat maps represent attention weights, with brighter colors indicating higher weights. Notably, the matrices are not symmetrical, reflecting the directional nature of attention —different attention is paid when "Madrid" is attending to "Spain" versus "Spain" to "Madrid". This characteristic reveals the quadratic complexity inherent in the attention mechanism, as each token's attention is calculated uniquely, leading to significant computational demands for longer sequences.

mechanism, encoder, decoder, residual connections, and layer normalization. Then, we will discuss the challenges and limitations of Transformers 2.2.5. Finally, we will explore recent advancements and the paradigm shift of Transformers 2.2.5 .

Understanding Attention Mechanism

Before we start covering the Attention mechanism, it is essential to note that Transformers and the Attention mechanism are not synonymous. Bahdanau et al. [82] in 2014, initially introduced the attention mechanism to address the bottleneck in RNNs handling tasks where the input and output sequence of a model have different lengths. But it was in 2017 that Vaswani et al. [6], in the paper "Attention is All You Need", introduced the Transformer model.

The Attention mechanism has been successfully integrated into other architectures like LSTMs [82] and CNNs [83], although not as a core feature. In contrast, the Transformer

architecture inherently incorporates and maximizes the potential of Attention, primarily through its multi-head self-attention mechanism, which we will explain later.

The Attention mechanism revolutionized natural language processing by addressing a fundamental limitation of recurrent models, which compress the entire input sequence into a fixed-length vector (the final hidden state), which can be a bottleneck for long sequences [84]–[86]. Consider the sentence "Madrid is a large city located in the centre of Spain". An RNN processes this sequentially, updating its hidden state one token at a time, failing to capture long-range dependencies between "Madrid" and "Spain" until the end of the sequence and not giving any importance for "Spain" when computing the hidden state of "Madrid" word. As we have mentioned in previous sections, to mitigate this and consider future elements in the sequence, the bidirectional RNN approach is required, which is computationally intensive.

On the other hand, **Attention mechanism** [6] resolves this by giving the model full access to the sequence of tokens when analyzing each token, allowing the model to consider all relationships within the sequence simultaneously and selectively focus on the most relevant information without compressing the context. For the same sentence, Attention can immediately relate "Madrid" to "Spain" by attending to both tokens simultaneously, **capturing the relationship regardless of their distance**, as evidenced by the high attention weights between these tokens in Figure 2.10a. The previous type of attention explained, is the Self-Attention also known as intra-attention, particularly crucial for Transformers, as we explain below.

Mathematical Formulation and Visual Representation

For a visual representation of the mathematical formulation you can check Figure 2.11. As depicted in this Figure, the attention mechanism involves three main components: queries (Q), keys (K), and values (V). These components are derived from the **first step** by multiplying the input embeddings (X) with trained weight matrices (W^Q, W^K, and W^V for queries, keys, and values, respectively). It is important to remark that, these three weight matrices are learnt by the model during the training process as a result of backpropagation, and the fact that each token in the input sequence has a corresponding q, k, and v vector associated with it. The process can be mathematically represented as follows:

$$\begin{aligned} \text{Queries } (Q) &= X \times W^Q \\ \text{Keys } (K) &= X \times W^K \\ \text{Values } (V) &= X \times W^V \end{aligned} \tag{2.4}$$

where:

- $X \in \mathbb{R}^{N \times d_{model}}$ is the input embeddings matrix, with N representing the number of tokens in the input sequence and d_{model} being the dimensionality of each token embedding
- $W^Q, W^K \in \mathbb{R}^{d_{model} \times d_k}$ are the weight matrices for queries and keys vectors
- $W^V \in \mathbb{R}^{d_{model} \times d_v}$ is the weight matrix for values vectors
- $Q, K \in \mathbb{R}^{N \times d_k}$ and $V \in \mathbb{R}^{N \times d_v}$

The **second step**, focus on understanding how important each token is for every individual token in the sequence. This is done by performing the dot product between the query vector of a given token (q_i) from the Q matrix and the key vector of all tokens individually (k_j) to compute a **score value** e_{q_i,k_j}, representing the relevance of each input token to the current token (2.5). Where q_i is the query vector for position i, k_j is the key vector for position j across the sequence length, l. For instance, if we're processing the self-attention for the token at position $i = 1$, the first score would be the dot product of q_1 and k_1, the second score would be the dot product of q_1 and k_2, and so on. This process can be interpreted as the token at position i asking other tokens how relevant they are to it by sending its query vector, and the other tokens responding with their key vectors, with the dot product between them quantifying their relevance.

$$e_{q_i,k_j} = q_i \cdot k_i \tag{2.5}$$

The **third step** is to divide the scores by the square root of the dimension of the key vectors ($\sqrt{d_k}$) to have more stable gradients, and then pass them through a softmax operation to normalize the scores, ensuring they are all positive and sum to 1, as shown in 2.6. This softmax score determines how much each input token will be expressed at the current position. The intuition here is that the higher the score for a token, the more relevant it is for the token whose representation is being computed. Clearly, the token itself at the current position will have the highest softmax score, but it's also useful to attend to other relevant tokens.

$$\alpha_{q_i,k_j} = softmax(e_{q_1,k_i}) \tag{2.6}$$

The **final step** (2.7) is to compute a weighted sum of the value vectors. For each value vector v_j, we multiply it by the corresponding **softmax score** α_{q_i,k_j}, which was computed between the query vector q_i and the key vector k_j of the same position j.

The intuition here is to keep intact the values of the relevant input tokens we want to focus on while diminishing the influence of irrelevant tokens. As a result, the tokens that are more relevant for the current position will have their values dominate the representation compared to less relevant tokens. Without the value vectors, the model would only have a linear measure of relevance (from the q-k comparison) but values allow the model to combine these vectors non-linearly increasing the knowledge of the model.

$$\sum_{j=0}^{l} \alpha_{q_i,k_j} \times v_j \tag{2.7}$$

We have shown the calculation for the self-attention focused on a single input token position i. However, in practice, we need to compute the self-attention for all token positions in the input sequence simultaneously. This open the doors for efficient parallel computation, where we can compute self-attention for all token positions simultaneously by condensing these steps into a single operation and using the Q, W and V matrices, obtaining the **general equation** (2.8), where the Attention output has $\mathbb{R}^{N \times d_v}$ dimensions.

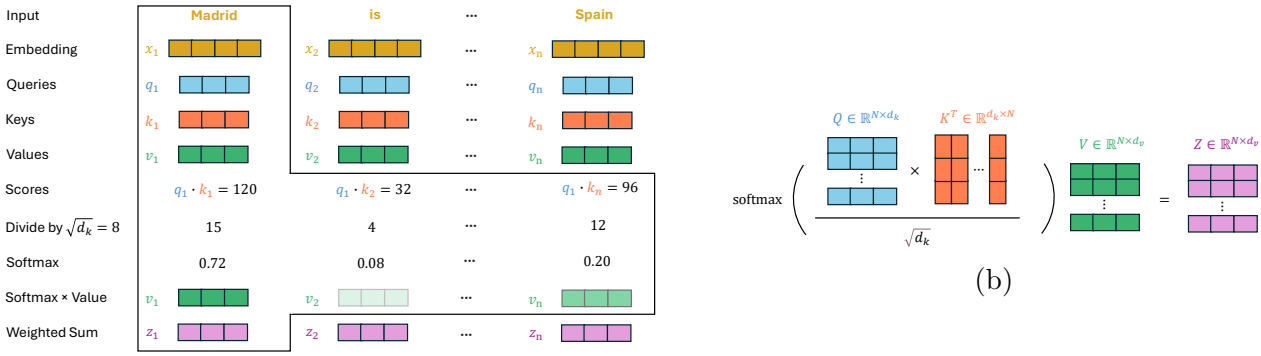


Figure 2.11: Visual Representation of mathematical formulation of self-attention. (a) Left panel breaks down self-attention for the input token "Madrid", showing the transformation into query, key, and value vectors, the calculation of attention scores, and the subsequent weighted summation to obtain the token's output embedding (z_1). Each step from scoring to final output highlights the attention's focus and how it varies for "Madrid" in relation to other tokens. (b) General matrix-level abstraction of self-attention, showing the condense computation for all tokens, illustrating the dimensions of query (Q), key (K), value (V), and output matrices (Z), essential for understanding the process's scalability and its quadratic computational complexity with respect to sequence length (N). (Adapted from: Jay Alammar jalammar.github.io).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.8}$$

So to recap, self-Attention compares all input sequence tokens with each other, and modifies the corresponding output sequence positions. In other words, self-attention layer differentiably key-value searches the input sequence for each inputs, and adds results to the output sequence with values. Perhaps some readers will find easier to think self-attention as a graph, especially (k-vertex) connected undirected (symmetric) weighted graph [87]. Self-attention is the core of the Transformer model, because not only enhances computational efficiency but also expands the model's ability to handle long-term dependencies understanding its relationship to every other element, regardless of their distance apart.

Multi-head Attention

The multi-head attention mechanism expands on this foundational concept by parallelizing the attention process by having different independents heads in on the same level, projecting the queries (Q), keys (K), and values (V) multiple times with different, learned linear transformations, as shown in Figure 2.12. This allows the model to capture information from different representational subspaces, not limiting the attention to be computed in one way but into multiple ways (heads). An analogy for this would be like working as a border security agent: Instead of having a single person analyzing a passenger from only one perspective, multi-head attention employs multiple experts (heads), each analyzing the passenger from a different viewpoint or with a different specialization, obtaining a much more rich analysis of

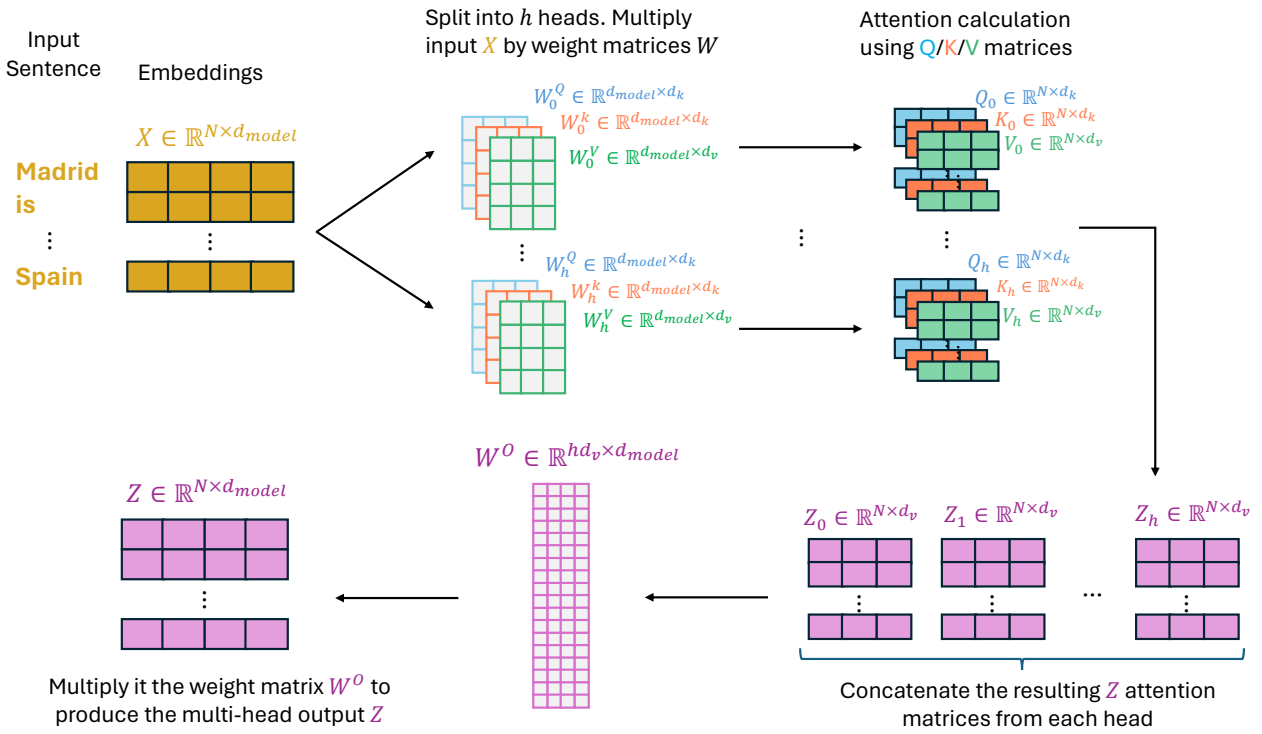


Figure 2.12: Visual Representation of Multi-head self-attention. (Adapted from: Jay Alammar jalammar.github.io).

the passenger (input sequence).

The multi-head attention can be mathematically represented as:

$$MultiHead\ Attention(Q, K, V) = Concat\,(Z_1, ..., Z_h)\,W^O = Z \tag{2.9}$$

Here, h represents the number of heads, and each head i produces an output Z_i defined by:

$$Z_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2.10}$$

The outputs (Z_i) of all heads are concatenated and then linearly transformed by W^O, producing the final output Z of the multi-head attention layer. To fully understand the operation of multi-head attention, we can check the dimensions of the projection matrices:

- Each Z_i operates in a dimension $\mathbb{R}^{N\times d_v}$, which concatenating h heads results in $\mathbb{R}^{N\times hd_v}$ dimensional space,
- W^O has dimensions $\mathbb{R}^{hd_v\times d_{model}}$ allowing the concatenated outputs of all heads to be transformed back into the model's original embedding space,
- The final output of the multi-head attention layer Z matches the original embedding dimension X input in the Attention layer, $\mathbb{R}^{N\times d_{model}}$. Therefore, the final output matches the input's dimensionality, maintaining consistency across the model's layers.

To put all this together, we can observe Figure 2.10 which represents two attention outputs from different layers of a Transformer BERT-base model[12] fine-tuned to detect Country and City names from a given sentence. Firstly, we can check that each attention matrix is different, but most importantly, within each matrix, several aspects are being studied. For example, in Layer 3, attention is paid to the relationship between a word and the next word, the relationship of all words with the [CLS] (sequence start) token, and the long-range relationship between "Madrid" and "Spain". In Layer 11, the relationship between "Madrid" and "Spain" is still captured, but the model also attends to the relationships between other words such as "centre", "city", and "located." This demonstrates how multi-head attention enables the model to focus on different aspects of the input sequence simultaneously, leading to a more comprehensive understanding of the contextual relationships.

With the provided explanation and visual representation (see Figure 2.12), we've demonstrated the consistency and utility of multi-head attention mechanisms within Transformers. Multi-head attention not only enhances the foundational attention mechanism by enabling concurrent processing across different representational subspaces, thereby endowing the Transformer with the capability to grasp a broader spectrum of linguistic relationships [84], [88]. This enhancement is significant not just because it allows the Transformer to focus on distinct facets of the input data simultaneously, but also because it leverages **parallelization** for efficiency. This is, partially but in big part, the success of Transformers and why we stated that Transformers include Attention as it cores.

Transformer Architecture

Overview Architecture

We have already discussed and explored the benefits of the self-attention mechanism. The Transformers' success is attributed to its neural network wrapper around the core concept of attention. It utilizes the self-attention mechanism in a sequence of encoders and decoders to exploit its benefits [85]. In Figure 2.13, we attach the original illustration of Transformers to enhance the explanation of Transformers architecture.

The encoder-decoder framework is at the heart of this architecture, where the encoder takes an input sequence and maps it to a sequence of continuous representations, which the decoder uses to generate an output sequence. The architecture can have several stacked encoder and decoder layers. The encoder layers include self-attention and feed-forward fully connected sub-layers, while the decoder layers include an additional attention sub-layer that interacts with the encoder output computing attention between encoder and decoder previous outputs.

Encoder

The encoder is the core component responsible for processing the input sequence and generating representations that capture its meaning and context. It is constructed as a series of identical blocks, each comprising two sub-layers:

- **multi-head self-attention** sub-layer

[12] https://huggingface.co/ml6team/bert-base-uncased-city-country-ner

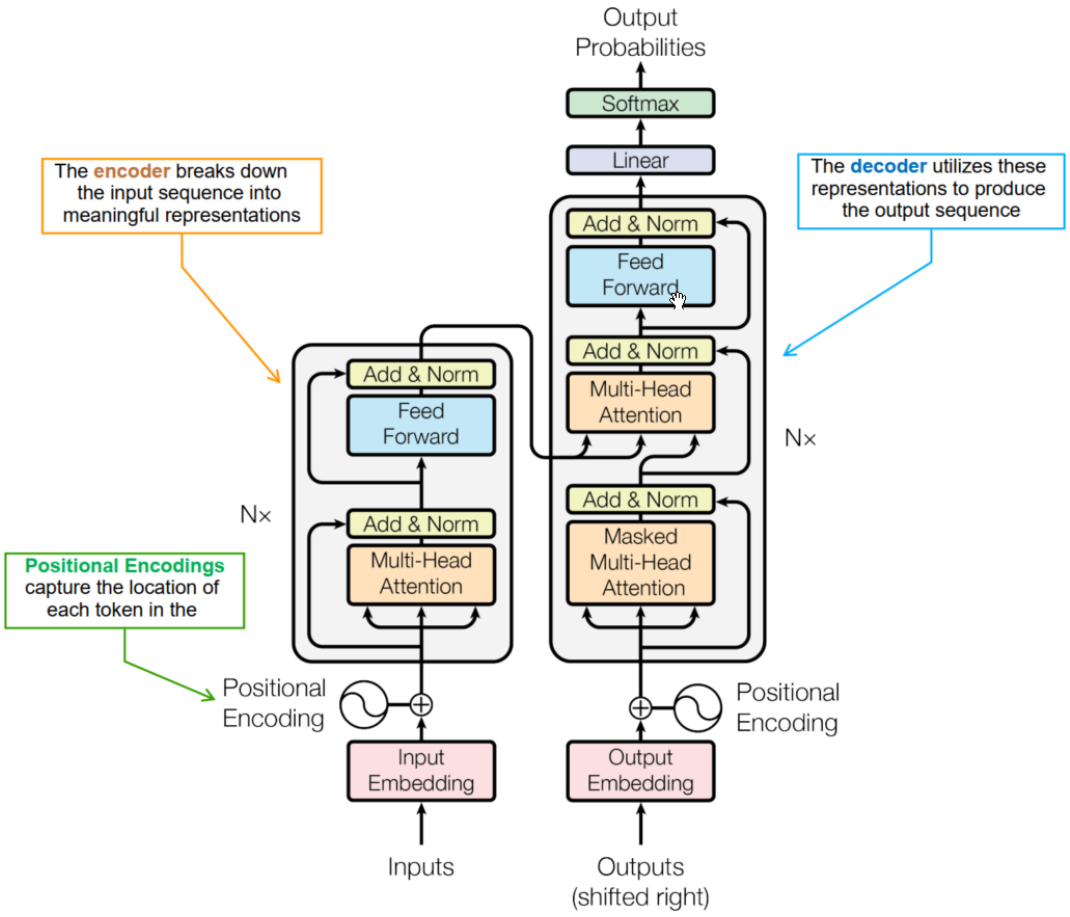


Figure 2.13: Transformer architecture proposed in "Attention is All you need" by Vaswani et al. [6]

- **fully connected feed-forward network (FFNN)**

As we have already explained, multi-head attention takes care of the parallel processing of the sequence and where the extraction of relationships among sequence elements takes place. The second sub-layer is a fully connected feed-forward network consisting of linear transformations with a ReLU activation function in between. This sub-layer refines the representations generated by the self-attention mechanism, further enhancing their expressiveness and capturing higher-level features.

Both sub-layers are wrapped in **residual connections** [89] and **normalization layers** [90], which play a crucial role in training deep neural networks effectively. The residual connections consist of adding to the output of the sub-layer its input, facilitating the flow of gradients during backpropagation, mitigating the vanishing gradient problem and enabling the construction of deeper architectures. Additionally, normalization layers normalize the residual sum of the sub-layer input and output, ensuring numerical stability and accelerating convergence.

One detail worth mentioning is that the Transformer cannot inherently determine the order of words in the input sequence due to its non-recurrent nature; remember that one advantage of attention is that all tokens are embedded considering the entire sequence of tokens, so

it is permutation invariant. This limitation is overcome by adding **positional encodings** to the input embeddings, which are embeddings with the same dimensionality as the input embedding from the tokenizer, to be fed to the Transformer. Various approaches exist for generating positional encodings. However, a standard method involves applying sine and cosine functions at varying frequencies to encode sequential information, indicating the tokens' absolute and relative position in the sequence.

Decoder

The decoder's role is to generate contextually output sequences from the linguistic information condensed by the encoder. Like the encoder, the decoder comprises identical stacked blocks, each comprising three sub-layers:

- a masked multi-head self-attention sub-layer
- a multi-head cross-attention sub-layer
- a fully connected feed-forward network (FFNN)

The decoder generates the output sequence in an **autoregressive** manner. This means that it processes the input sequence token by token, generating the output sequence one token at a time.

The **masked multi-head self-attention sub-layer** operates similarly to the encoder's self-attention mechanism, enabling the decoder to capture dependencies within the partially generated output sequence. At any given position, the masked multi-head self-attention sub-layer can only attend to the previously generated tokens, capturing dependencies within this partially generated output sequence. Hence, it employs a masking technique to prevent attending to future positions, ensuring that predictions for a specific position depend solely on the known outputs from preceding positions.

The decoder's **fully connected feed-forward network sublayer** serves a similar purpose to its counterpart in the encoder, refining and enriching the representations generated by the attention sublayers. These sublayers also include residual and normalization layers.

The **multi-head cross-attention sub-layer** is a unique decoder component that enables it to attend to the representations generated by the encoder. Similar to the self-attention mechanism, this sub-layer computes attention weights, but instead of attending to the decoder's own representations, it attends to the encoder's output representations. This allows the decoder to capture the relationships between the input and output sequences generated, which is crucial for tasks like machine translation, where the output is conditioned on the input.

General Purpose NLP model

The Transformer architecture's encoder and decoder components are designed to play distinct yet complementary roles. The encoder processes the input sequence, generating a set of continuous embeddings representations that capture its meaning and context. On the other hand, the decoder generates the output sequence based on these representations from the encoder, attending to both the input sequence and the partially generated output sequence.

This is why transformers, although initially designed for machine translation, are widely considered general-purpose NLP models, as they can be adapted to various configurations based on the task [51], [85]. An encoder-only configuration can be employed for tasks like text classification and named entity recognition, where the encoder generates representations of the input sequence. In contrast, a decoder-only configuration is used for tasks like text generation, next-word prediction, and fill-in-the-blanks, relying solely on the decoder component. Finally, both encoder-decoder components are utilized for sequence-to-sequence tasks such as machine translation or text summarization.

For the **scope of this Thesis**, we focus on the encoding part for extracting most of the linguistic characteristics from text into latent space and using it for developing multilingual solutions for fighting misinformation. The decoder generation capabilities are less focused in this Thesis because generating content is not the approach proposed to combat misinformation. However, it was necessary to mention the decoder to explain the attention mechanism and the characteristics of decoders, which makes clear why models like the GPT decoder-only family models, have breakthroughs in the AI field, that we encourage interested readers to consult [91]. Nevertheless, we will now see some challenges and limitations of these architectures.

Challenge and Limitation

Despite their remarkable success, Transformer models face several inherent limitations that restrict their applicability and efficiency, particularly when dealing with extensive contexts or resource-constrained scenarios [88]. These challenges include:

One of the primary limitations of Transformer models is their **fixed-length context window** [86]. Vanilla Transformers are constrained by a predefined maximum sequence length due to their self-attention mechanism, which requires quadratic memory complexity with respect to the sequence length. This limitation becomes a significant bottleneck when processing long documents, as their ability to model relationships over longer distances beyond the fixed context size is limited. As the input length grows, the self-attention mechanism becomes a computational bottleneck, hindering scalability.

Closely tied to the fixed-length context window is the high computational complexity associated with the self-attention mechanism. For a sequence of length N, the model considers N-to-N relationships, causing the **computational complexity to increase quadratically** as $O(N^2 d)$ where d denotes the model's hidden dimension. It has been shown that these quadratic cost of self-attention impacts speed in both training and inference [88]. On the other hand, the complexity og feed-forward layers at every Transformer block is linear with respect to the sequence length N. Notably, early models were restricted to a context length of 512 tokens, but advancements in computing have pushed attainable attention increase to between 2K, 32K or more tokens [92]. However, scalability issues persist under the vanilla Attention framework, with the quadratic growth of matrix sizes requiring overwhelming GPU memory capacities, thereby imposing physical constraints on context lengths.

While Transformers are capable of learning powerful representations, their efficiency in doing so, especially with limited data or in unsupervised scenarios, can be improved [84]. Their success **rely on training over vast amounts of data and computational resources**, so

pre-training Transformer models from scratch can be prohibitive for organizations with limited computing resources. Attention mechanism is a double-edge sword, because its high flexibility assumes minimal structural bias in the input data While this flexibility is advantageous in many cases, it can also make them challenging to train effectively on small-scale datasets [93]. Hence, for tasks with limited labeled data, achieving good performance can be difficult from scratch. This requires for more data-efficient learning mechanisms to make Transformer models more accessible and applicable in resource-constrained settings.

Despite their advancements over RNNs and GRUs, **vanilla Transformers do not incorporate inherent recurrence mechanisms** [94]. As previously mentioned, they rely on positional encodings to model sequence information, which may not fully capture temporal dependencies. While this may not be a significant limitation for many natural language processing tasks, but it is worth noting that for tasks requiring strong temporal modeling, the absence of recurrence can be a potential drawback.

These limitations have motivated ongoing research efforts to address and mitigate these challenges, aiming to enhance the scalability, efficiency, and applicability of Transformer models in various domains and resource-constrained settings. In the following sections, we will explore some of the research carried out and ongoing work to address these limitations.

Recent Advancements and Future Directions

The landscape of Natural Language Processing (NLP) models, particularly Transformers, is continuously evolving. This section presents the recent advancements made to address some of the limitations previously discussed, as well as the promising directions for future research and concepts that are exploited and applied in the scope of this Thesis.

As previously mentioned, Transformers, by their very design, face challenges related to the fixed-length context window and computational complexity [88]. Initially, researchers sought to mitigate these issues by making the attention matrix sparse [95]. This was achieved by limiting the scope of attention to fixed, predefined patterns, such as local sliding windows that move along sequences with overlapping strides. A notable advancement came with the Longformer model [96], which utilizes a sliding window mechanism. This mechanism restricts each token to attend only to nearby tokens, along with a few global tokens for broader context, effectively reducing computational complexity to $O(N\sqrt{N}d$, a significant improvement. Furthermore, other research [97] has focused on low-rank approximations of the self-attention matrix, offering another avenue for efficiency based on the assumption that a product of two smaller matrices can approximate the attention matrix. The Linformer [98] model exemplifies this approach by projecting the keys and values to a lower-dimensional space, easing the memory complexity inherent to self-attention as the $N \times N$ matrix is now decomposed to $N \times k$, reducing the self-attention complexity to $O(N)$ in both time and space.

Downsampling the sequence's resolution also emerged as a method to decrease computation [88]. Downsampling can consist, for example, of selecting a subset of tokens from the original sequence or combining multiple tokens into a single representation to reduce computation costs [99]. Nevertheless, these approaches have a clear trade-off between efficiency and model performance, a dilemma that remains a focal point for ongoing research.

Model Category	Examples	Advantages Use Cases	Limitations Challenges
Statistical NLP	GMMs HMMs CRFs TextTiling	• Good for structured data. • Low computational cost.	• Lacks context awareness • Assumes word independence • Limited by annotated data and rules • Difficulty in capturing non-linear relationships.
Traditional Context-Independent	BoW+NB TF-IDF+SVM	• Simple to implement. • Effective for text classification with clear boundaries	• Ignores word order, context, and polysemy • Decoupled tokenization and vectorization • Fixed word representations • Sparsity • Not suitable for large vocabularies • Limited adaptability to new, unseen data.
Advanced Context-Independent	Word2Vec Gensim FastText	• Integration of tokenization and vectorization • Captures semantic similarity of words • Manages to handle rare words.	• Lacks context awareness and polysemy • Fixed word representations
Hierarchically Structured Models	DBN DNNs	• Model complex patterns • Effective for feature extraction • Effective for generative modeling	• Requires large datasets • Computationally intensive • Additional steps required for supervised tasks (DBNs) • Potential overfitting with very deep networks
Spatial Hierarchy	CNN	• Captures local dependencies • Useful for text classification	• Fixed Input Size • Limited contextual understanding • Not optimal for sequence-to-sequence tasks • Requires filter tuning
Sequence Modeling	RNN	• Variable input and output sizes • Exploit sequential data • Contextual Integration	• Vanishing/exploding gradient problems • Limited memory span
Gated Sequential Models	LSTM GRU BiLTSM	• Handles Long-Term Dependencies • Mitigates the vanishing gradient problem • Process variable-length sequences effectively.	• More computationally intensive than RNNs • Selective memory • Heavy training/tuning.
Attention Models: Transformers	BERT RoBERTa GPT Transformer-XL	• Long-Term Dependencies without recurrence • Self-attention mechanisms to weigh the importance. • Distance-Agnostic Relationship modeling • Parallel processing with multi-head attention rather than sequentially. • Parallelizable • General purpose (encoder-only, decoder-only, encoder-decoder). • Pre-train and Fine-Tune paradigm	• Fixed context window • Quadratic complexity • Bottleneck scalability for long sequences • Large data/compute requirements for training from scratch. • Lack of Recurrence, requires positional encodings. • Lack of Interpretability

Table 2.2: A concise overview of various model categories in Natural Language Processing (NLP), including their examples, advantages/use cases, and limitations/challenges.

The paradigm of pre-training and fine-tuning was brought up regarding the significant data and computing requirements for training and developing an effective Transformer model [100], [101]. This methodology allows for the exploitation of models in downstream tasks without necessitating training from scratch. Additionally, it reduces the amount of labelled data, which is expensive and time-consuming. This approach underscores the potential of transfer learning, where knowledge from one task can significantly benefit a different but related task. Then, fine-tuning the model in a downstream task enhances target task performance. Such strategies have democratized sophisticated models, making them accessible without extensive data or computational resources. Nevertheless, with the breakthrough of Large Language Models (LLMs), even the pre-trained version of these models is prohibitively large enough to be fine-tuned, presenting new gaps [85], [88].

In pursuit of more efficient models, bi-encoders [102], [103] have been proposed as an alternative to the initial Transformer's encoder, also named cross-encoders, for sentence similarity. Cross-encoders perform full self-attention between the two sentences. Bi-encoders, on the other hand, perform self-attention separately on the input sentences, mapping each of these to a dense vector space and then combining them for a final representation, enabling faster prediction times and efficient indexing, compared to the slower inference speeds of cross-encoders.

A well-known technique but novelty applied in neural networks is quantization [104]. The quantization technique consists of representing continuous data (like real numbers) in a discrete form and, for example, rounding 32-bit Floating Points (Single Precision) to 6-bit Floating Points (Half Precision) or to 8-bit Fixed Point integers. This technique has been proposed for neural networks [105], [106], reducing the memory footprint and computational demands of Transformer models, enabling their deployment even in resource-constrained environments. Similarly, BitNet [107], [108] from Microsoft introduces a model with ternary parameters (-1, 0, 1, challenging traditional notions by maintaining performance with significantly reduced GPU memory and energy consumption.

Another limitation previously mentioned came from the double sword of not applying recurrence in Transformers. Transformers initially compensated for their lack of recurrence with positional encodings, but newer research has introduced more sophisticated methods. For instance, Transformer-XL [86] incorporates segment-level recurrence and relative sinusoidal embeddings to extend context beyond fixed lengths. The Rotatory Positional Encoding (RoPE) [109] method innovatively combines absolute position and relative distances through a rotational mechanism in the embedding space. This ongoing research is focused on offering a novel solution to encode positional information effectively for better handling long dependencies and improving how the model extracts the relationship between elements in a sequence.

The scalability of Transformers has been significantly bolstered by industry investments since 2014. This is shown with the release and the race of releasing groundbreaking Large Language Models like GPT-4 [110] from OpenAI, Gemini [111] from Google, Claude[13] from Anthropic, or Llama [112] from Meta, which belong to the decoder-only Transformer family, evidencing how Transformers is the backbone of many cutting-edge applications [113], [114].

[13] https://www.anthropic.com/news/claude-3-family

These models underscore how important attention is and their widespread adoption across various domains beyond NLP, such as computer vision and speech recognition. Models like Gemini claim to be multi-modal thanks to cross-modal attention, which is based on the idea of applying attention to data from different modalities (e.g., text and audio). This underscores the transformative impact of attention mechanisms, proving their applicability beyond mere text analysis.

In conclusion, while the original Transformer model has revolutionized NLP and other areas, the identified limitations have sparked a wave of innovation. Addressing computational complexity, enhancing model efficiency, and expanding the model's applicability across domains is necessary; the future of Transformer technology appears both promising and boundless. As research continues to evolve, we can anticipate further breakthroughs that will push the boundaries of what these powerful models can achieve.

2.2.6 Multilingualism in NLP

Motivation and Importance of Multilingual NLP

Human language is inherently diverse, with over 7,000 languages spoken globally, each with its unique structure, vocabulary, and nuances as reported by recognized institutions Ethnologue[14] and Instituto Cervantes [115]. As the world becomes increasingly interconnected, the ability to process and understand this linguistic diversity through computational models is of paramount importance.

The widespread adoption of the Internet and social media has accelerated global communication, transcending geographic boundaries. In 2024, a staggering 5.35 billion people, or 66.2% of the world's population, are using the Internet [116]. Moreover, there are now over 5 billion active social media users worldwide, with 266 million new users joining in 2023 alone, an average of 8.4 new users per second. This combined with the fact that new generation are increasingly using social media for information rather than traditional communication platforms remarks the growing demand for technologies that can bridge language barriers [19].

Multilingual NLP solutions have emerged as a critical frontier in computational linguistics, aiming to develop Transformer models that can understand and process multiple languages simultaneously [51], [117]. This field seeks to address the inherent diversity of human language and democratize technology, making it accessible to all, regardless of the language spoken.

Challenges in Developing Multilingual NLP Models

While multilingual transformer models represent a breakthrough, developing effective solutions faces several significant challenges [51], [117], as described below.

One of the primary obstacles is **data representation and preprocessing**. Transforming textual data from diverse languages into a suitable numeric representation that can be effectively processed by the models is a non-trivial task. Tokenization (as explained in Section 1), is a crucial early step in any natural language processing system, presents its

[14] https://www.ethnologue.com/insights/how-many-languages/

own set of difficulties when dealing with low-resource languages that have limited data availability [118]. Early studies [119], [120], have highlighted the importance of employing appropriate tokenization algorithms to manage multiple languages within a single model.

Alongside tokenization challenges, there is a critical trade-off to be addressed between **vocabulary size and vocabulary entropy**, which directly impacts the model's performance and computational efficiency [52], [118]. For better understand this **trade-off**, vocabulary size refers to the total number of unique tokens (words, subwords, or characters) in a given language, where a larger vocabulary size allows the model to represent a broader range of linguistic units with less out-of-vocabulary (OOV) issues but requires more memory and computational resources [66], [67]. On the other hand, vocabulary entropy measures the uncertainty associated with predicting the next token given the context. In the context of multilingual Transformers, achieving an optimal balance means ensuring the model's vocabulary is comprehensive enough to cover the diversity of languages without becoming so large that it dilutes the specificity and predictability of language features [121].

Secondly, **different languages** have vastly **different structures**, vocabularies, and word orders, making it difficult to develop a single, universal model that can effectively capture and process these variations [115]. Nevertheless, training separate models for each language is a resource-intensive and inefficient approach, necessitating the development of a single, multilingual model capable of processing text in multiple languages simultaneously while dealing with this diversity [103], [117].

Another significant challenge lies in the **skewed availability of online content and resources across languages**. According to W3Techs data, over 63% of the world's top 10 million websites have content in English, increasing since 2021 [122]. While languages like Russian and Turkish have a significant online presence, others like Chinese and Spanish, despite having large populations of speakers, are relatively underrepresented on the web. As of 2023, the dominance of English persists, with 52% of online content in English, followed by Spanish, German, and Russian at 4-5% each [116]. However, this distribution is gradually shifting, further underscoring the importance of developing multilingual NLP models that can adapt to the evolving linguistic landscape of the Internet and global communication. For any interested reader, the statistics and further analysis of this data can be accessed at DataReportal Reports[15][16].

As depicted in Figure 2.14, languages are classified as **high, medium, or low resource**, depending on the computational data resources available [117], [123]. This uneven distribution of resources across languages compounds the challenges associated with tokenization, data representation, and linguistic diversity, necessitating innovative solutions to develop robust, equitable, and representative multilingual models.

In the previous section we have described the functioning of Transformers models and their great potential in exploiting the attention mechanism. Multilingual Transformers models are still Transformers but capable of processing, representing in a dimensional space different languages to condense the linguistic features into a continuous numerical embedding

[15] https://datareportal.com/reports/digital-2024-global-overview-report
[16] https://datareportal.com/reports/?tag=Global+Overview

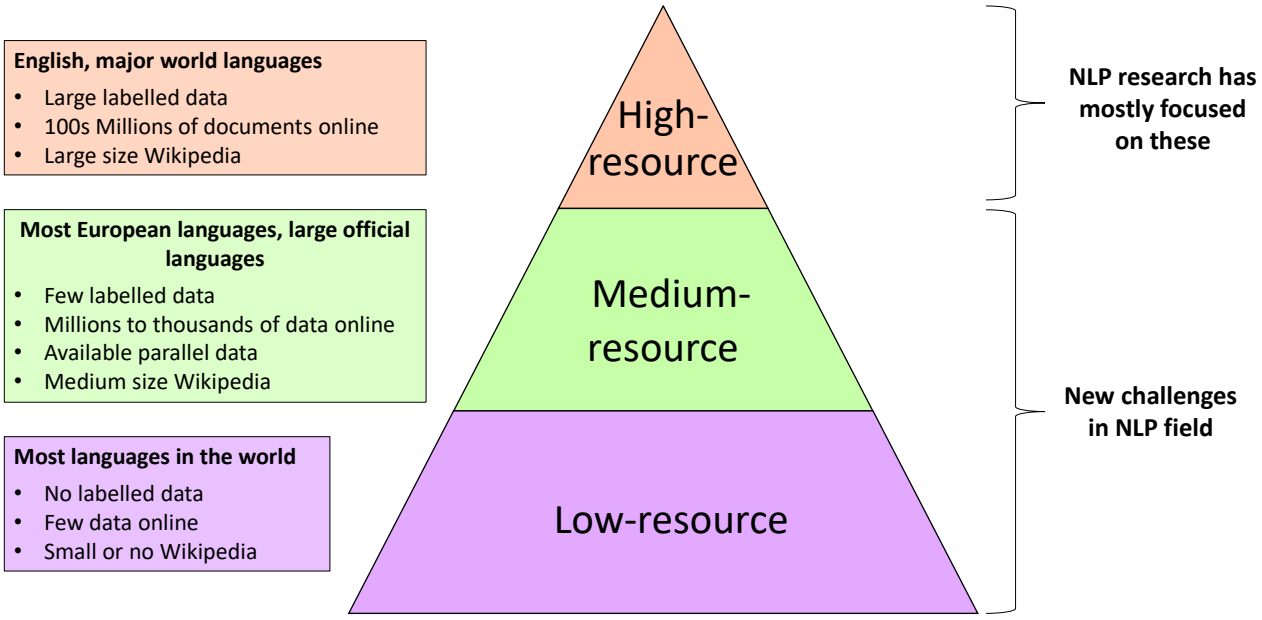


Figure 2.14: Conceptual view of the Natural Language Processing resource hierarchy. Note that many languages cannot be assigned clearly to a single level of the hierarchy. Adapted from: [117].

representation that can be used for an NLP goal or task. So the architecture shares the same elements, with details to adapt and allow them to incorporate multiple languages.

Multilingual Transformer models represent a significant advancement in NLP, designed to understand and generate multiple languages within a single framework. These models are built on the Transformer architecture, leveraging its powerful self-attention mechanism to process text from different languages simultaneously [51], [91], [124]. The core idea is to create a shared representation space where languages coexist, enabling the model to transfer knowledge across languages. Focusing in multilingual Transformers not only advance the state-of-the-art in NLP technologies but also democratize access to information across language barriers, promoting inclusivity and facilitating cross-cultural communication

Tokenization for Multilingual Language Representation

Tokenization plays a pivotal role in multilingual Transformers, as these models are designed to simultaneously handle and process text from multiple languages [118], [121], [125]. To achieve this, multilingual Transformers rely on a shared vocabulary that can accommodate the linguistic features of all the languages they are trained on. Tokenization is the **crucial initial step** in constructing this shared vocabulary, breaking down text from various languages into manageable units—tokens—that the model can understand and process.

As discussed in Section 2.2.3, tokenizers can be organized into four categories [52]: Word-Level Tokenization, Morphological-Level Tokenization, Character-Level Tokenization and Sub-word level tokenization. A visual example of the tokenization output of these different categories is shown in Table 2.1.

NLP applications initially relied on word-level tokenizers, which divided text into words using spaces and punctuation. While producing fewer unique vocabulary items, word-level tokenizers could not effectively handle out-of-vocabulary (OOV) words [121]. NLP techniques and models already presented in previous Section 2.2.4, such as the Bag-of-Words [56] model and Word2Vec [61], utilized this type of word-level tokenization. Additionally, Character-level tokenizers have been used in CNNs and RNNs models [126], [127], but operating directly on individual characters, treating each character as a token has no significant benefit but heavier computation associated, increasing pattern for the performance of all tokenizers as the vocabulary size increases.

In contrast, subword-level tokenization has emerged as a hybrid, more effective approach [118], [128]. It significantly reduces the occurrence of OOV units and enhances the model's adaptability to new vocabularies and complex language phenomena. This approach offers several advantages, firstly, considering text from a language-agnostic perspective and reducing the overall vocabulary size while still capturing meaningful linguistic information.

Algorithm 1 Algorithm for Tokenization using Byte Pair Encoding. Adapted from [53]

Input: A corpus of text, desired vocabulary size V_f
Output: A sub-word vocabulary

Step 1: Initialize vocabulary V_i with unique characters in the corpus
Step 2: Tokenize the corpus into a sequence of characters, appending each word with a special end symbol </w> along with their frequencies

while size of $V_i < V_f$ AND pairs to merge exist **do**
  Step 3: Identify the most frequent pair of (b_1, b_2) in the corpus.
  Step 4: Merge b_1 and b_2 into a new token t_{new}
  Step 5: Replace occurrence of (b_1, b_2) with t_{new} in the corpus
  Step 6: Recalculate frequency of all tokens, considering updated corpus from Step 5
  Step 7: Add t_{new} to the vocabulary V_i
end while
Step 8: Return the final sub-word vocabulary.

The development of subword tokenization algorithms has been driven by several key advancements:

- FastText [Bojanowski et al., 2017]: Laid the groundwork for subword-based tokenization by considering subword information. Unlike traditional word embeddings in Word2Vec, FastText breaks down words into smaller units (subwords or n-grams) such as character sequences. This approach allows FastText to effectively handle out-of-vocabulary (OOV) words by using subword representations. Additionally, FastText's subword-based approach makes it language-agnostic, enabling its application across multiple languages. Even though it was primarily designed for word embeddings, FastText's idea and contribution influenced subsequent tokenization methods that Trnasformers exploited.
- Byte-Pair Encoding (BPE) [53]: Introduced in 1994 for data compression [130], BPE

Algorithm 2 Algorithm for Tokenization using WordPiece. Adapted from [129]

Input: A corpus of text, desired vocabulary size V_f
Output: A sub-word vocabulary

Step 1: Initialize vocabulary V_i with unique characters in the corpus
Step 2: Tokenize the corpus into a sequence of characters, appending each word with a special end symbol </w> along with their frequencies

while size of $V_i < V_f$ **do**
Step 3: Evaluate each possible new token combination (b_x, b_y) from V_i for likelihood gain if added to V_i
Step 4: Select the token t_{new} that maximizes the likelihood increase of training data
Step 5: Replace occurrence of selected (b_x, b_y) with t_{new} in the corpus
Step 6: Recalculate likelihood of all tokens, considering updated corpus from Step 5
Step 7: Add t_{new} to the vocabulary V_i
end while
Step 8: Return the final sub-word vocabulary.

gained prominence in the NLP community in 2016 due to its effectiveness in handling subword tokenization for neural machine translation [53]. BPE follows a greedy strategy (see Alg.1, starting with a vocabulary of individual characters or subwords. It then iteratively merges the most frequent character pairs to create new subword units, effectively reducing the vocabulary size. Frequent symbol pairs are replaced with a special token (e.g., @@) to represent the merged subword. The merging process continues until a predefined vocabulary size is reached. BPE significantly reduces the vocabulary size, making it more efficient for processing large-scale language data, and can represent languages with rich morphology using a compact vocabulary. Its success is evident in its adoption for GPT-2 [131] and subsequent GPT models by Radford et al. [132] in 2020.

- WordPiece [129], [133]: WordPiece is another subword tokenization algorithm, closely related to BPE but with a slight difference in the merging criteria (see Alg. 2. Like BPE, WordPiece merges subword units, but it selects the pair that maximizes the likelihood of the training data when added to the vocabulary, rather than merging the most frequent token bigram. WordPiece starts with a base vocabulary and iteratively merges pairs to enhance representation. WordPiece is used in multilingual Transformers (such as XLM [124]) to handle diverse languages, balancing tokens and types (unique vocabulary items), enhancing model adaptability while controlling complexity. Multilingual Transformers leverage WordPiece to create embeddings that capture cross-lingual semantics. Its success is highlighted by its adoption in the influential BERT model by Devlin et al. [133] in 2017.

- SentencePiece [134] and Unigram models [135]: While BPE and WordPiece focus on specific merging criteria, SentencePiece, introduced in 2018 [134], aims for flexibility and language independence. For sake of clarity, Unigram is the method of obtaining subword vocabulary used as the underlying tokenizer of SentencePiece algorithm [136]. It is

designed to strength the language-independent subword tokenizer that treats the entire training corpus as a single sentence and learns subword units (e.g., subwords, characters) based on likelihood (see Alg. 3). The Unigram model optimizes tokenization based on likelihood, like WordPiece, but takes a different approach. SentencePiece combines the strengths of BPE (frequency-based) and Unigram (likelihood-based) models, offering subword regularization and adaptability across languages. Hence, it can handle both word and subword units, offering a fast C++ implementation of BPE introducing likelihood-based tokenization and subword regularization, leading to more robust and semantically meaningful representations, especially in multilingual settings. It has been successfully applied in multilingual versions of models such as BART [137] and T5 [138], [139].

Algorithm 3 Algorithm for Tokenization using Unigram Language Model. Adapted from: [135]

Input: A corpus of text, desired vocabulary size V_f
Output: A sub-word vocabulary

Step 1: Initialize vocabulary V_i with unique characters and all possible subwords from the corpus.
Step 2: Initialize probabilities for each sub-word based on their occurrence in the corpus.

while size of $V_i > V_f$ AND significant changes in vocabulary can still be made **do**
Step 3: Optimize the probability of each sub-word by maximizing the likelihood of the corpus under the current vocabulary.
Step 4: Compute the loss for each sub-word, quantifying the impact of its removal on the overall corpus likelihood.
Step 5: Sort sub-words by their loss and retain the top $X\%$ to form the new, reduced vocabulary, ensuring character-level sub-words remain to prevent out-of-vocabulary issues.
Step 6: Update V_i by removing the least important sub-words as determined in Step 5.
Step 7: Recalculate the probabilities for the remaining sub-words in the newly pruned vocabulary.
end while
Step 8: Return the optimized sub-word vocabulary, now pruned to size V_f or optimized for corpus likelihood.

These advancements in subword tokenization algorithms have played a crucial role in the development of multilingual Transformer models, enabling effective language modeling across diverse linguistic contexts [128].

When choosing a tokenization method for multilingual Transformer models, considerations such as linguistic richness, computational efficiency, and the specific application scope must be taken into account [52], [118]. WordPiece, for instance, provides better linguistic richness due to its likelihood-based approach, while BPE is computationally efficient and widely adopted. Additionally, there is ongoing research focused on improving tokenization techniques for low-resource languages and enhancing cross-lingual transfer capabilities to leverage knowledge

from high-resource languages effectively [121], [136].

These advancements in subword tokenization, have significantly improved the potential of multilingual Transformers to create shared vocabularies and representations that accommodate the nuances of multiple languages, leveraging the power of the Transformer architecture and self-attention mechanism to learn universal language features and facilitate cross-lingual understanding and transfer.

This tokenizer step is the crucial initial step for the subsequent processes of shared representation space construction during pre-training, and fine-tuning for downstream tasks, which will be discussed in the following sections.

Pre-training Multilingual Transformer Models

After tokenization, the model is trained to learn shared representations across languages through self-supervised pre-training.

We have already mentioned the powerful **"Pre-train and Fine-tune Paradigm"** [100], [101] approach for developing Transformer models. This pre-training consists of self-supervised learning (SSL) [88], [91], which belongs to unsupervised learning, where the model creates its own supervision from the data. SSL pretrains models at scale based on pseudo-supervision offered by one or more pre-training tasks. The pseudo-supervision stems from automatically generated labels without human intervention, based on the description of the pre-training task. We will cover widely used and crucial pre-training SSL tasks for multilingual Transformers.

To better understand the pre-training and fine-tuning stages, consider how humans learn a new language in a language academy (pre-training) and then adapt that foundation to their daily lives (fine-tuning). Firstly, humans undergo initial exercises such as filling sentence gaps with vocabulary, learning phrasal verbs, and distinguishing between tenses and irregular forms. Once we have learned the basics and gained control over the language, we fine-tune our knowledge by adapting our vocabulary and expression to specific domains in our daily lives. Just like a medical doctor would use different terminology than an aerospace engineer, we adjust our language skills to suit different contexts.

Multilingual Pre-training Corpora

It is important to note the significance of data before exploring pre-training techniques. A diverse and representative pre-training corpus is critical for pre-training multilingual transformer models. This corpus must cover a wide range of languages, including high-resource and low-resource, to ensure broad linguistic representation. Some of the most relevant multilingual pre-training datasets in the literature include Multilingual Common Crawl (mC4, CC25) extracted from the Common Crawl web archive [140], [141], the Parallel Sentences Corpus [142], which contains sentences translated into various languages for cross-lingual representations, and the CLARIN and OPUS Projects [143], which have parallel corpora.

However, a still common challenge lies in the varying representativeness of languages in the training data. This imbalance can affect how the tokenizers and the transformer models represent the texts in the shared representation space [117], [141]. Thus, carefully curating the

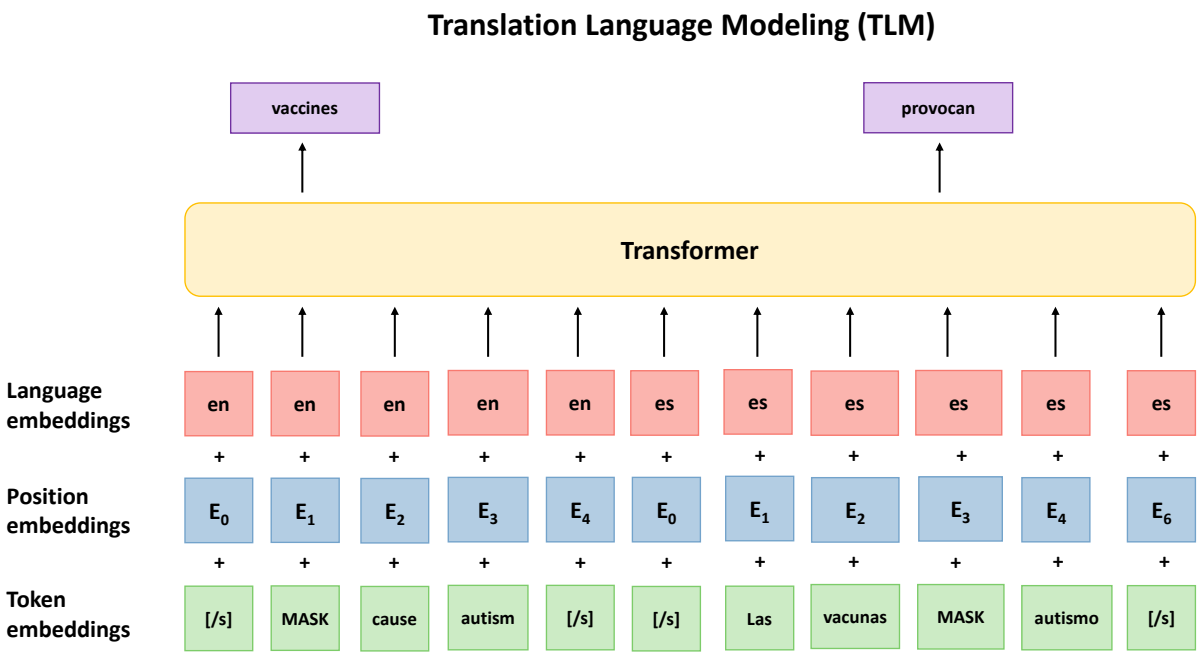


Figure 2.15: Translation Language Modeling (TLM) pre-training task visualization from XLM multilingual Transformer pre-training. Position embeddings of the target sentence are reset to facilitate the alignment. Adapted from [124]

pre-training corpus is crucial to mitigate potential biases and ensure equitable representation of diverse linguistic phenomena.

Techniques for Effective Multilingual Pre-training

In addition to a diverse and carefully selected pre-training corpus, the pre-training techniques used are crucial for enabling multilingual transformer models to learn shared representations across languages effectively.

These techniques aim to enhance aspects of multilingual language understanding, such as language identification by incorporating language tags that helps the model recognize and adapt to the input language [144]. Also, creating scenarios where the model learns to handle language-specific characteristics, addressing language-specific traits like syntax, morphology, and semantics across diverse linguistic contexts [145]. Furthermore, these pre-training tasks should be challenging enough to allow the model to learn semantics at different levels (word, phrase, sentence, or document), while providing sufficient training signal for the model to acquire more language information with less pre-training data. Depending on the desired output, it may also be beneficial to design pre-training tasks that closely align with downstream tasks, reducing the gap between pre-training and fine-tuning [120].

Several techniques have been developed to address these aspects not just in monolingual but also in multilingual Transformer models pre-training:

- **Masked Language Modeling (MLM)**: Introduced by BERT (Bidirectional Encoder Representations from Transformers) [133], MLM randomly masks a percentage of the input tokens. The model's objective is to predict these masked tokens based on their

context, encouraging a deep understanding of language context and word relationships. Extensions of MLM include Random Token Substitution (RTS), where tokens are randomly replaced but never with a special MASK token, and Aggregated Probabilities of Token Misclassification (e.g., Cluster-based Random Token Substitution, C-RTS), focusing on discriminating original tokens from substituted ones [144]. Another variation, the Swapped Language Model (SLM), is trained to predict the original tokens, providing more training signal compared to MLM, as it covers all input tokens rather than a subset, effectively reducing pre-training discrepancy [120], [144].

- **Next Sentence Prediction (NSP)**: Also introduced by BERT [133], NSP involves processing pairs of sentences and predicting whether the second sentence is the subsequent sentence in the original document. This fosters learning sentence-level semantics, including topic and coherence prediction. However, due to the simpler nature of topic prediction, the effectiveness of NSP has been debated. A related task, Sentence Order Prediction (SOP), focuses solely on sentence coherence to increase difficulty and enhance learning, employed by models like BERT and ELECTRA [146].
- **Causal Language Modeling (CLM)** [147]: Utilized by models such as GPT (Generative Pre-trained Transformer) [131], CLM involves predicting the next token in a sequence given the preceding tokens, training the model to understand language sequentially from left to right, which is beneficial for generative tasks.
- **Translation Language Modeling (TLM)** [124]: TLM is also referred to as cross-lingual MLM, since it input is a pair of parallel sentences (same sentence each in a different language), and tokens from both sentences are masked (see Figure 2.15). Essential for multilingual models like mBERT [133] and XLM [141], TLM trains on parallel texts in various languages, promoting the cross-lingual mapping alignment of different languages' representations at the token level. This approach is especially valuable for tasks requiring cross-lingual understanding or translation.
- **Contrastive Learning** [148]: This technique improves learning by contrasting positive examples (e.g., correct translations or paraphrases) against negative examples (e.g., incorrect translations). Although used in models like CLIP [149] for aligning text and image representations, it's adaptable for purely textual tasks to enhance understanding of language differences. This pre-training task is especially relevant in the scope of multilingual Transformers since it requires the models to learn shared representations across languages [120]. As a result, the model is forced to align linguistic representations across languages in the same shared space, making it easier for the model to transfer knowledge from one language to another, which is particularly beneficial for low-resource languages where direct training data might be scarce.
- **Denoising Pre-training**: Techniques like sentence permutation [150] and span corruption [138] serve as effective denoising pre-training objectives for robust multilingual pre-training, focusing on reconstructing logically coherent text or correct spans from corrupted input. These techniques are beneficial for learning robust multilingual representations, and can be complemented by language-specific pre-training tasks, like TLM. It has been utilized utilized by models like multilingual BART [150], which learns to

reconstruct the original text from corrupted versions, training encoder-decoder-based models on the task of text restoration.

- **Translation Pre-training** [124]: This bilingual pre-training strategy use sentence-level translation pairs for training models to translate sentences from one language to another. It improves the ability of multilingual models to handle translation tasks effectively. It has been successfully applied to enhance the performance of mT5 multilingual and cross-lingual tasks [139], [151].
- **Object Entailment Pre-training Task** [152]: Designed to enrich the model's understanding of structured relationships, given a subject and relation, the model predicts the object from a list of candidates, thereby learning to infer connections within data. This kind of techniques are closer to downstream tasks but are useful since models are forced to develop a deeper understanding of language structure and semantics.

Fine-tuning and Adaptation

After the pre-training stage, where the multilingual Transformer model learns shared representations across languages from a carefully curated multilingual corpus, the next step is fine-tuning and adaptation. Pre-training allows the language model to gain universal language knowledge; however, to achieve optimal performance on specific downstream tasks, the model's weights need to be adapted to acquire task-specific knowledge at specific downstream tasks, languages, or domains [120], [153].

Fine-tuning imparts this task-specific knowledge by adjusting the model's weights based on the target task's loss function [133]. This process enhances the model's performance by increasing the separation between clusters of different labels, ensuring better discrimination and generalization [154]. Interestingly, during fine-tuning, the higher layers of the Transformer model undergo more substantial changes compared to the lower layers [154], [155], allowing for effective specialization while retaining the universal language knowledge gained during pre-training.

Researchers and practitioners can leverage well-known pre-trained multilingual models, such as XLM (based on Byte-Pair Encoding) [124], XLM-RoBERTa (SentencePiece), multilingual BERT (WordPiece) [133], mBART-50 (SentencePiece) [150], BLOOM (byte-level BPE), and cutting edge models Gemma (byte-level BPE) [111], LLama (byte-level BPE) [156], and fine-tune them for specific downstream tasks and domains.

The pre-trained multilingual model serves as a strong starting point for fine-tuning on a wide range of downstream tasks, such as translation, text classification, named entity recognition, and question answering. Fine-tuning the pre-trained model on task-specific data allows for customization and optimization of the model's performance across multiple languages.

Some commonly used multilingual datasets for fine-tuning include [145] :

- XNLI (Natural Language Inference): The XNLI [157] is a popular evaluation dataset for cross-lingual NLI which contains 15 languages.
- XQuAD [158] (Question Answering): IT is a benchmark dataset for evaluating cross-

lingual question answering performance composed of paragrpahs and pairs of question-answer pair with the corresponding translation version.

- XTREME [159]: XTREME covers 40 languages maimizing language diversity consisting in 9 NLP tasks that require reasoning about different levels of syntax or semantics.
- MLQA [160] (Question Answering): It is a multilingual question answering dataset, which covers 7 languages including English, Spanish, German, Arabic, Hindi, Vietnamese and Chinese.

Following this fine-tuning approach, the multilingual Transformer can leverage its learned shared representations and adapt them to the nuances of the target task, achieving high performance across languages it was not explicitly trained on during pre-training [120].

In scenarios where optimal performance in a specific language is essential, language-specific fine-tuning can further enhance the model's effectiveness [154]. This approach involves fine-tuning the pre-trained multilingual model using data from the target language, allowing the model to specialize in that language's details while retaining its cross-lingual capabilities.

A cornerstone of multilingual Transformers is their ability to facilitate cross-lingual transfer learning [124], [141], [148]. During fine-tuning, the model can leverage its shared representations to transfer knowledge from high-resource languages, where abundant data is available, to low-resource languages with limited data. This cross-lingual transfer capability is crucial for handling the world's linguistic diversity, enabling the model to perform well even on languages with scarce training resources.

By leveraging these pre-trained multilingual models and following the pre-training and fine-tuning paradigm, facilitated by self-supervised learning techniques and careful data curation, researchers and practitioners can achieve state-of-the-art performance on a wide range of multilingual tasks, demonstrating the versatility and effectiveness of multilingual Transformers in bridging linguistic barriers.

Challenges and Future Directions

Despite the impressive capabilities of multilingual Transformers, research continues to explore their limitations and work towards overcoming challenges such as resource-lean scenarios, language distance, and the need for more effective cross-lingual transfer methodologies [120].

As previously mentioned in Subsection 2.2.6 about the challenges of developing multilingual Transformers compared with monolingual models, one significant challenge is the scarcity of data for the majority of the world's languages [118].

One solution to bridge the data gap is to employ translation-based data, which is indeed helpful in leveraging inter-lingual capabilities. Many existing multilingual datasets, like XNLI [157], XTREME [159], utilize this approach. However, real-world data becomes crucial to capture the authentic linguistic characteristics of target languages. This can be achieved through unsupervised and self-supervised learning techniques that exploit the abundant unlabeled data available on the web for low-resource languages [161]. Additionally, synthetic data generation emerges as a potent solution, enriching the diversity and volume of training data, which also

have collateral benefits safeguarding privacy, increasing data availability for research, and reducing bias in machine learning models [162].

Beyond data scarcity, applicability for underrepresented languages also faces limited computing resources in these countries [163]. This challenge extends beyond processing power to include access to mobile data and other technological infrastructures needed to collect and generate well-curated data, often costly or nonexistent in regions that speak these languages [161], [164]. For this reason, strategies for model compression, quantization, and knowledge distillation are vital. They can facilitate the deployment of state-of-the-art models on devices with limited computing capabilities.

Additionally, as data are scarce for training, there is a lack of evaluation benchmarks to properly assess and track progress in multilingual capabilities. This issue is aligned with a broader challenge across AI, where the HAI 2023 report [1] identified that improvements of models on well-established benchmarks are marginal, indicating saturation of these benchmarks. Thus, establishing comprehensive benchmarks to measure model performance across languages and tasks accurately is essential [165].

The future of multilingual NLP depends on addressing these challenges through innovative research and community collaboration. Areas such as unsupervised learning, the strategic use of synthetic data, cross-lingual transfer learning, and multimodal integration promise to unlock new potentials in NLP applications, making language technologies more inclusive and accessible across the globe. Furthermore, the exploration of universal models that include different languages and modalities without extensive resource requirements will continue to be a key focus. The aim is to bridge the gap between high-resource and low-resource language scenarios, ensuring equitable advancements in NLP.

2.3 Literature Review on Computational Approaches against False Information

As we have already explained in Section 2.1, the proliferation of false information has become an urgent global challenge, particularly on social media platforms. This literature review provides a broad overview of computational approaches, emphasizing natural language processing (NLP) techniques, to address this issue. However, numerous strategies can be employed, given the multidisciplinary nature of false information detection and mitigation.

The following chapters and publications of this Thesis will present the specific background and related works most relevant to the explored approaches and techniques. Each publication will review the pertinent state-of-the-art and prior art to contextualize the novel contributions to tackling various aspects of the false information challenge. Building on the concepts and definitions covered earlier, this literature review provides a broad overview of historical development.

Computational Approaches

Early efforts in computational approaches to false information primarily focused on rule-based systems and traditional machine learning methods. These techniques relied on manually crafted rules or features derived from the text to classify information as true or false. For example, researchers explored linguistic features, such as the presence of specific words or phrases, to identify deceptive language [166].

As the field of computational approaches to false information detection progressed, it witnessed a shift from rule-based systems and traditional machine learning methods to more sophisticated techniques. These new methods, such as supervised machine learning algorithms like support vector machines (SVMs) and logistic regression, were able to learn patterns from labelled data [167], [168]. They leveraged features like text content, user credibility, and propagation patterns to detect false information. However, these approaches were limited by their reliance on manually engineered features and the availability of labelled data.

The advent of deep learning and neural networks revolutionized computational approaches to false information, particularly in NLP. These techniques enabled the automatic extraction of high-level features from raw text data, alleviating the need for manual feature engineering.

One approach involved using recurrent neural networks (RNNs) and long short-term memory (LSTM) networks to capture the sequential nature of text and model the temporal dynamics of information propagation [169]. These models could learn representations of text and user behaviour to identify patterns indicative of false information.

With the rise of transformer models, such as BERT [133] and RoBERTa [170], pre-trained language models (PLMs) have become a driving force in NLP tasks, including false information detection. These models, trained on large corpora of text data, can capture rich semantic and contextual information, enabling effective transfer learning for downstream tasks. Their ability to understand the nuances of language and context has significantly improved the accuracy and efficiency of false information detection systems [171], [172].

To address this limitation, researchers have developed social media-specific transformer models by either pretraining from scratch on large social media corpora, like BERTweet [173], or through continual pretraining of existing models on social media data [172], [174]. Some models, like BERT-SentiX [175], have even explored novel pre-training tasks tailored for social media content. To address this limitation, researchers have developed social media-specific transformer models by either pre-training from scratch on large social media corpora, like BERTweet, or through continual pre-training of existing models on social media data [174].

Ensemble methods that combine news content analysis with social network features have emerged as one of the most effective strategies for fake news detection [176]. However, challenges persist regarding the generalizability, explainability, and bias in these approaches [177]. Deep learning techniques, such as attention mechanisms, and generative adversarial networks, offer advanced capabilities in detecting false information more accurately than traditional machine learning methods [120]. While current methods perform well within restricted domains, generalized fact-checking can benefit from the integration of powerful Internet search engines.

Moreover, researchers have explored modeling the spread of false information, incorporating

mechanisms like fact-checking, varying forgetting behavior, and different underlying network structures [12]. These efforts aim to understand and potentially mitigate the propagation dynamics of misinformation and disinformation across social networks and online platforms.

Multimodal Approaches

While NLP techniques have been prominent, researchers have also explored multimodal approaches that incorporate additional modalities, such as images, videos, and social network metadata, to enhance the detection of false information. These multimodal approaches aim to leverage the complementary information provided by different data sources to improve the overall performance of false information detection systems, including images, videos, and audio.

For instance, researchers have explored the similarity between textual content and visual information [178] or using visual features extracted from multiple images [177]. Additionally, studies have investigated the relevance of metadata, such as user credibility, propagation patterns, and temporal dynamics, in identifying false information [172].

Challenges and Opportunities

Despite significant advancements, false information detection and mitigation remain challenging due to the dynamic and adversarial nature of this problem. Key challenges include deep learning models' lack of explainability, the labour-intensive and time-consuming process of manual fact-checking, and the co-evolution of malicious agents and detection solutions (content evasion) [12], [120], [177].

Moreover, the deficit of multilingual datasets and the bias toward English in existing models and datasets leave non-English-speaking regions more vulnerable to misinformation (Alam et al., 2021). Addressing this gap by developing multilingual datasets and models is crucial for a more inclusive and practical approach to false information detection. Another significant challenge lies in the temporal dynamics of information veracity, as the truthfulness of a news item may change over time as new information becomes available [179]. Additionally, examining diverse data modalities, such as images, videos, and social network metadata, and developing robust models capable of synthesizing insights from these disparate data sources remains a non-trivial problem [12].

Despite these challenges, computational approaches to false information present numerous opportunities for further research and innovation. Advancements in transformer architectures, multilingual language models, and multimodal fusion techniques commit to more robust and practical solutions. Furthermore, integrating domain-specific knowledge, such as fact-checking databases and expert-curated knowledge graphs, could enhance these systems' reasoning capabilities.

Chapter 3

Methodology

3.1 Pre-selection of Domain and Case Studies

This section provides an overview of the main focus areas of this thesis, which are centered around combating false information through advanced computational techniques. The research is structured around three key pillars, visually presented in Figure 3.1:

- dimensionality reduction in multilingual contexts
- content evasion detection,
- robustness against evasion techniques.

These pillars directly address the research questions outlined in Section 1.3, focusing on leveraging Natural Language Processing (NLP) and Deep Learning to tackle the complex challenges related to false information in the digital age.

3.1.1 False Information Domain

This thesis is dedicated to exploring computational strategies, particularly through applying Natural Language Processing (NLP) techniques, to help counter the escalating issue of false information, including misinformation and disinformation. As stated in the Motivation Section 1.3, the spread of false information across online platforms and social media represents a substantial global challenge and carries profound societal, political, and economic impacts, as extensively discussed in literature review on false information (Chapter 2.1).

The domain of false information presents unique challenges for computational approaches. This problematic often manifest in various forms such as fake news articles, manipulated images or videos, coordinated social media campaigns, and more. The technical challenges in this domain include: (1) High volume and velocity of data generation and spread, (2) Multimodal nature of false information (text, images, videos), (3) Linguistic complexity and cultural nuances (4) Evolving tactics of malicious actors to evade detection.

While multimodal approaches has been explored in associated congress publications [180]

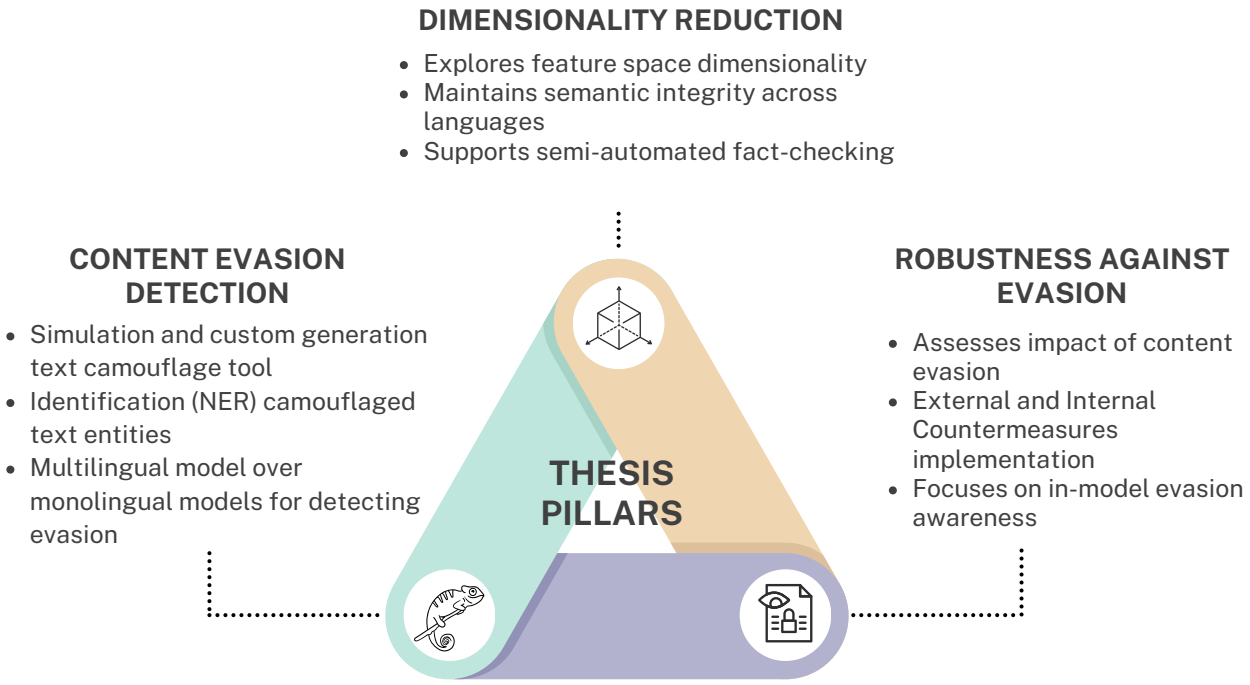


Figure 3.1: An overview of the pillars of the thesis.

at SemEval-2022 Multimedia Automatic Misogyny Identification (MAMI) task, the thesis research focuses primarily on textual false information, leveraging NLP techniques to address these challenges.

Given the critical importance of this false information problem domain, this methodology section outlines the key focus areas and case studies that have been selected to guide the research presented in this thesis. The thesis aims to develop robust NLP-based solutions to detect, mitigate, and counteract the spread of false information across diverse online platforms and languages. This aligns directly with the research questions focusing on utilizing computational tools, specifically within the realms of NLP and Deep Learning, to address the complex and evolving landscape of false information.

3.1.2 Overview of Research Approach

This thesis employs a comprehensive methodology to identify challenges and gaps in the state of the art, which guided the research conducted in the publications comprising this thesis by compendium. These publications collectively form the pillars of the thesis.

The general methodology used to identify these challenges and opportunities for advancing beyond the state of the art is as follows:

- Comprehensive Literature Review: An extensive analysis of current research in false information detection, NLP techniques, and deep learning models was conducted. This first step of reviewing helped identify key challenges and research gaps across multiple dimensions of the false information problem.

- Gaps and Challenges Identification: The identified challenges were mapped to specific areas within NLP and deep learning where current techniques fall short in addressing false information problematic, particularly in multilingual contexts and against evolving evasion tactics.
- Hypothesis Formulation: Based on the identified gaps, hypotheses were formulated about potential improvements in NLP techniques to combat false information more effectively.
- Pillar Development: The research was structured around three key pillars previously mentioned, each addressing a critical aspect of the false information challenge: a. Dimensionality reduction in multilingual semantics b. Content evasion detection through camouflage identification c. Enhancing robustness against evasion in Transformer models
- Iterative Research and Development: Each pillar was developed through cycles of model development, experimentation, and evaluation, with continuous refinement based on results.
- Validation and Dissemination: Results from each study were validated against state-of-the-art benchmarks and disseminated through academic publications, which now form the compendium of this thesis.

This methodological approach allowed for a systematic exploration of the complex challenges in false information detection and mitigation, while maintaining a focus on advancing the state of the art in NLP applications.

3.1.3 Research Flow and Stages

The research progressed through several interconnected stages, each building upon the previous:

- Initial Exploration: The dimensionality reduction study laid the groundwork by optimizing the feature space of multilingual Transformer models.
- Application to Specific Challenges: Insights from the first study informed the development of techniques for content evasion detection, particularly in identifying camouflaged text.
- Robustness Enhancement: The final study focused on improving model resilience against adversarial attacks, incorporating learnings from both previous stages.

This flow ensured that each research pillar contributed to a comprehensive approach to combating misinformation using advanced NLP techniques. Below, each pillar and the context of a case study for each one is described in more detail.

3.1.4 Dimensionality Reduction in Multilingual Semantics

There are several language levels, including phonetics, phonology, morphology, syntax, semantics, and pragmatics. Each level plays a crucial role in the comprehension and generation of language, with phonetics and phonology focusing on sounds, morphology and syntax on the

structure of words and sentences, semantics on the meaning conveyed by words and phrases, and pragmatics on language use within context [44].

The decision to focus on semantics in this thesis is not arbitrary. It is driven by its fundamental importance in natural language processing (NLP) and the inherent challenges it presents, particularly in the context of multilingual applications and relevance in the domain of false information. Semantics, the study of meaning, is at the core of human communication, interpreting words, phrases, sentences, and entire texts to grasp the intended meaning. This is crucial for the research questions addressing the use of Transformer models to identify and interpret nuanced linguistic patterns essential in the fight against misinformation.

Addressing multilingual semantics is both a choice and a necessity, driven by the complexity of comparing and aligning meanings across languages [51], [161]. Given the diversity and intricacy of linguistic expressions worldwide, this task presents considerable challenges. However, multilingual transformer models, such as those developed and evaluated in the study, are designed to understand and process multiple languages, making them suitable for tasks that involve semantic comparison of texts in diverse linguistic contexts. Thus, the investigation of these models within this thesis is motivated by their potential to address the semantic challenges inherent in multilingual NLP applications.

Furthermore, this research also examines the dimensionality reduction techniques and their implications for Semantic Textual Similarity (STS) tasks. Dimensionality reduction aims to compress high-dimensional data, such as word or sentence embeddings, into a more compact representation while preserving the essential semantic information. Achieving this balance is crucial, as the loss of semantic content can significantly reduce the performance of downstream NLP tasks that rely on these compact representations. This directly supports the thesis's goal of optimizing and fine-tuning the feature space of Transformer models to enhance their ability to identify and interpret nuanced linguistic patterns.

Moreover, models that effectively capture and represent semantic details facilitate a more accurate understanding of the text, which can be exploited to discern truthfulness better and integrated into broader systems designed to detect and mitigate misinformation, as proved in a collaboration article.

3.1.5 Content Evasion Detection through Camouflage Identification

The second case study within this thesis focuses on the evasion of content moderation through word camouflage techniques [181].

Content evasion refers to the tactics and strategies employed by individuals or entities to bypass the content moderation systems implemented by online platforms. These moderation systems filter out prohibited, harmful, or misleading content based on predefined rules and guidelines. However, malicious actors have developed various evasion techniques, such as the use of leetspeak (replacing letters with visually similar numbers or symbols), insertion of unnecessary punctuation, or word inversion, to alter the content in a way that it avoids detection by automated moderation tools while remaining readable and understandable to human users (see Table 3.1).

Case of study	Original	Camouflaged	Source
Gaming	noobs	n00bz	Blashki et al. 2005
	owned	pwn3d	
	skiils	sk11lz	
	fear	ph34r	
Password	HBOpassword	#B0p4$$w0r|)	Hong et al. 2021
Cybersquatting	incibe.es	incive.es	Instituto Nacional de Ciberseguridad (INCIBE)
		incIbe.es	
		inci-be.es	
		inicbe.es	
Social Media COVID-19 Infodemic	vacuna	v4cun4	EU Disinfolab
		b4cun4	
		v@(u¬a	
		nacuva	
		V.A.C.U.N.A	
	covid	k0 b1t	
		K0b1d	
		c0*vid	
		C(o(v(i(d	

Table 3.1: Examples of camouflage technique applied in different situations, as shown in real-world cases documented by previous studies and references. These examples illustrate the diverse and creative ways in which malicious actors may modify text using Leetspeak to evade content moderation.

Countering these evasion techniques is extremely important for several reasons. Maintaining platform integrity is crucial, as evasion techniques can undermine the efforts of online platforms to create safe and respectful environments for users by allowing harmful content, such as hate speech, misinformation, or cyberbullying, to persist undetected. Protecting users, particularly vulnerable groups, from exposure to such content is a primary goal of content moderation, making effective evasion countermeasures essential.

Applying natural language processing (NLP) techniques offer a promising solution to this challenge. NLP enables the automation of detecting and understanding complex text patterns that human annotators might find challenging to process at scale. NLP's ability to understand the context and semantics of text is crucial for identifying subtle forms of misinformation or content evasion that might not be apparent through surface-level analysis. This aligns with the thesis's motivation to leverage computational techniques to assist human-supervised activities and improve procedures to keep pace with the rapidly evolving nature of false information.

3.1.6 Enhancing Robustness Against Evasion in Transformer Models

This thesis's third central focus area investigates the robustness of Transformer models against adversarial text attacks, specifically through content camouflage techniques. This exploration is critical to understanding and enhancing the resilience of NLP systems deployed to combat false information, aligning with the overarching thesis motivation of addressing the challenges posed by misinformation and disinformation.

Adversarial attacks, where visually altered texts deceive AI models, pose a significant threat to the integrity of automated misinformation detection systems. Such attacks exploit the vulnerabilities in NLP models, leading to incorrect processing and outputs. This research evaluates how different Transformer configurations, such as encoder-decoder, encoder-only, and decoder-only models, handle these sophisticated adversarial strategies. The relevance of this study to the thesis motivation lies in its focus on developing resilient NLP models capable of withstanding adversarial manipulation. This aligns with the research question of optimizing Transformer models to identify better and interpret nuanced linguistic patterns essential in the fight against misinformation. By understanding and countering these adversarial techniques, the thesis creates more robust models that maintain high accuracy and reliability in real-world scenarios.

Furthermore, the study's emphasis on external countermeasures, including MASK and BLANK filters and the introduction of static and dynamic adversarial training methods, showcases innovative approaches to enhance model robustness. These methods not only mitigate immediate adversarial impacts and serve as potent data augmentation techniques, improving overall model performance. This directly supports the thesis goal of leveraging computational tools to develop effective solutions for the rapidly evolving landscape of false information.

By integrating this research into the broader narrative of the thesis, which includes false information, semantics, and dimensionality reduction, the focus on adversarial attacks ensures a comprehensive approach to understanding and mitigating the spread of misinformation. This alignment with the thesis's core objectives underscores the necessity for resilient AI systems that can adapt to and counter sophisticated adversarial challenges, thereby reinforcing the commitment to developing robust NLP solutions to combat false information across diverse platforms and languages.

3.2 Computational Resources

Here are listed the computational resources used in this Thesis:

- Operating System: Ubuntu 20.04 LTS (Focal Fossa) and 22.04.4 LTS (Jammy Jellyfish) x86_64 architecture.
- CPU: Intel(R) Xeon(R) Bronze 3206R CPU @ 1.90GHz with 8 cores.
- RAM: Total 252GB of RAM available
- Storage: Type HDD, 9TB capacity.

- GPU: Quadro RTX 8000 48 GB. Given its ability to concurrently manage thousands of threads, the GPU was employed primarily for tasks necessitating high parallelism, such as the training of deep learning models.
- Python Version 3.9 and 3.10 [182] was the primary programming language used to conduct the research. Pip and Conda were the package and virtual environments managers employed.

Specialized Tools and Frameworks

The following are the mainly package and tools used in this Thesis for coding experiments and visualize and share results:

- Scikit-learn [183]: A popular Python library for machine learning. This library was employed to load algorithms and implement dimensionality reduction techniques considered in the study, along with metrics for NLP model performance measurement.
- PyTorch [184] (v2) and PyTorch Lightning [1] (v2): Frameworks for developing NLP models, with PyTorch Lightning acting as an extension to streamline dataloader, training, and evaluation function development.
- Hugging Face Transformers [113]: A cornerstone library for NLP used to load pre-trained models and provide a framework to perform tasks on texts.
- Hugging Face Tokenizers [185] and SentencePiece [134]: Libraries designed to handle tokenization, transforming raw text into a format suitable for machine learning model inputs. Additionally to the use for training models, these were used to explore the impact of camouflage on tokenization algorithms.
- Sentence Transformers [102]: A Python framework for state-of-the-art sentence embeddings, used for training Transformer networks fine-tuned for semantic similarity.
- Emoji[2]: A library for preprocessing text and converting Unicode emojis to and from emoji names before feeding the Transformer models.
- SpaCy [186]: An open-source library for advanced natural language processing in Python, used to preprocess and format data for NER tasks and develop multilingual NER models.
- Scipy [187]: An open-source Python library for mathematics, science, and engineering, used here for statistical computations.
- UMAP [188]: A library implementing Uniform Manifold Approximation and Projection for dimensionality reduction. Addtiionally, Pooling and Numba are used to complement UMAP. Numba is a just-in-time compiler for Python, and pooling to manage memory and parallel processing tasks.
- GitHub [189]: A platform for version control and collaboration, used here to share reproducible results.

[1] https://zenodo.org/record/3828935
[2] https://pypi.org/project/emoji/

- AugLy[text] [190]: A library from Facebook for introducing modifications in texts, used as an external validation tool for camouflage experiments.
- CodeTiming[3]: A flexible Python library for timing code execution, useful for decorators and context managers.
- Yake [191]: A Python library for unsupervised automatic keyword extraction, used for comparing keyword extraction using semantic-aware models.
- KeyBert [192]: A library that exploits semantic-context aware text embeddings to extract keywords closely associated with a text.
- NLTK [193]: A software library for natural language processing in Python, used for extracting punctuation and stopwords.
- Unidecode[4]: A library used to process text in different formats as ASCII to handle camouflage versions of texts properly.
- Pyphen[5]: A module for hyphenating text using existing dictionaries, supporting multilingual text.
- CuPy [194]: A library for GPU-accelerated computing with Python, used to enhance the performance of SpaCy tools on GPUs.
- For data and results visualizations Plotly [195], Matplotlib [196] and seaborn [197] are used, and Streamlit[6] to turns Python scripts into shareable web apps in minutes, used for rapid prototyping and data science.

3.3 Multilingual Transformers Models

3.3.1 Relevance of Transformers

As we have discussed in Section 2.2.5, it is clear that Transformers use an attention mechanism that allows them to focus selectively on certain parts of the input sequence, making it easier to understand the meaning of a sentence. Additionally, transformers are based on a self-attention mechanism that allows them to process the entire input sequence in parallel, significantly increasing computational efficiency. Transformers are the state-of-the art and for this we focus our case of studies using this architecture.

In that Section, we have explained that Transformer are models that can be applied in different task with different configurations: encoder-only, decoder-only and encoder-decoder. For the scope of this Thesis, we focus on the encoding part for extracting most of the linguistic characteristics from text into latent space and using it for developing multilingual solutions for fighting misinformation. The decoder generation capabilities are less focused in this Thesis because generating content is not the approach proposed to combat misinformation.

[3] https://pypi.org/project/codetiming/
[4] https://pypi.org/project/Unidecode/
[5] https://pypi.org/project/pyphen/
[6] https://docs.streamlit.io/

3.3.2 Knowledge Distillation

As discussed in Section 2.2.6, developing effective multilingual NLP models faces several significant challenges, including issues with data representation, linguistic diversity, and the uneven distribution of resources across languages. To address these limitations, this Thesis employs pre-trained multilingual models developed using knowledge distillation techniques, initially introduced by Bucila et al. [198] and later generalized by Hinton et al [199].

The core idea behind knowledge distillation is to leverage the capabilities of a high-performing, monolingual "teacher" model to train a multilingual "student" model [120], [144]. This technique is illustrated in Figure 3.2. The teacher model, pre-trained on large, high-resource datasets, has acquired a deep understanding of language structure and semantics. By training the student model to mimic the behavior of the teacher, the linguistic knowledge can be transferred to the multilingual model, allowing the student to benefit from the teacher's insights without the need for extensive training on diverse language data.

In this thesis, we utilize pre-trained multilingual models that have been developed using knowledge distillation and are publicly available through the Hugging Face Transformers library. Specifically, we make use of multilingual models from Sentence Transformer which have been trained using knowledge distillation techniques to create high-performing, cross-lingual representations [103].

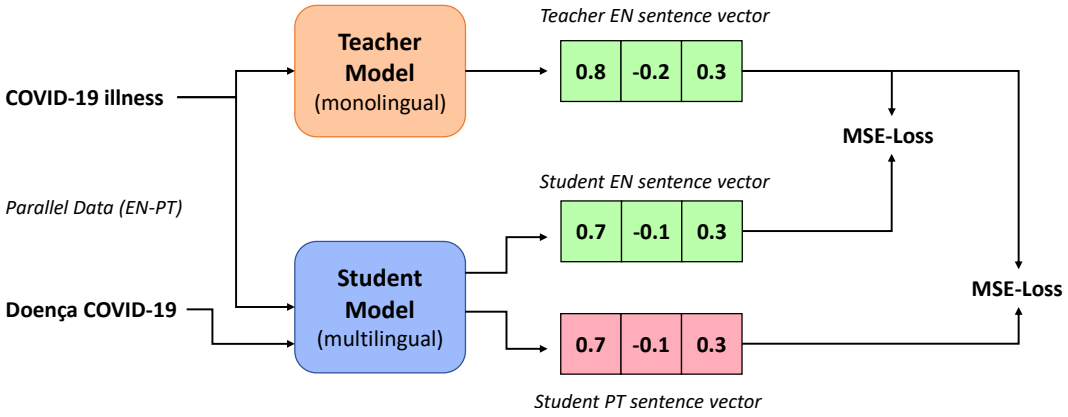


Figure 3.2: Given parallel data (e.g. English and Portuguese), train the student model such that the produced vectors for the English and German sentences are close to the teacher English sentence vector. The multilingual student model imitates the teacher model and achieves by this a high performance. Adapted from [103].

3.3.3 Bi-encoders

In addition to leveraging pre-trained multilingual models developed through knowledge distillation, this thesis also explores the use of bi-encoder Transformer architectures. Bi-encoder models, also known as dual-encoder or siamese networks, are a specific configuration of Transformer-based models that are particularly well-suited for tasks involving semantic similarity comparison, such as the semantic textual similarity (STS) task [102], [200].

As illustrated in Figure 3.3, siamese architectures consists of two pre-trained Transformer-based models with tied weights that can be fine-tuned on a specific task like compute similarity scores. The key distinguishing feature of bi-encoder architectures is that they employ two separate Transformer encoder networks to process the two input texts independently. This contrasts with single-encoder models, where a single Transformer is used to encode both inputs simultaneously.

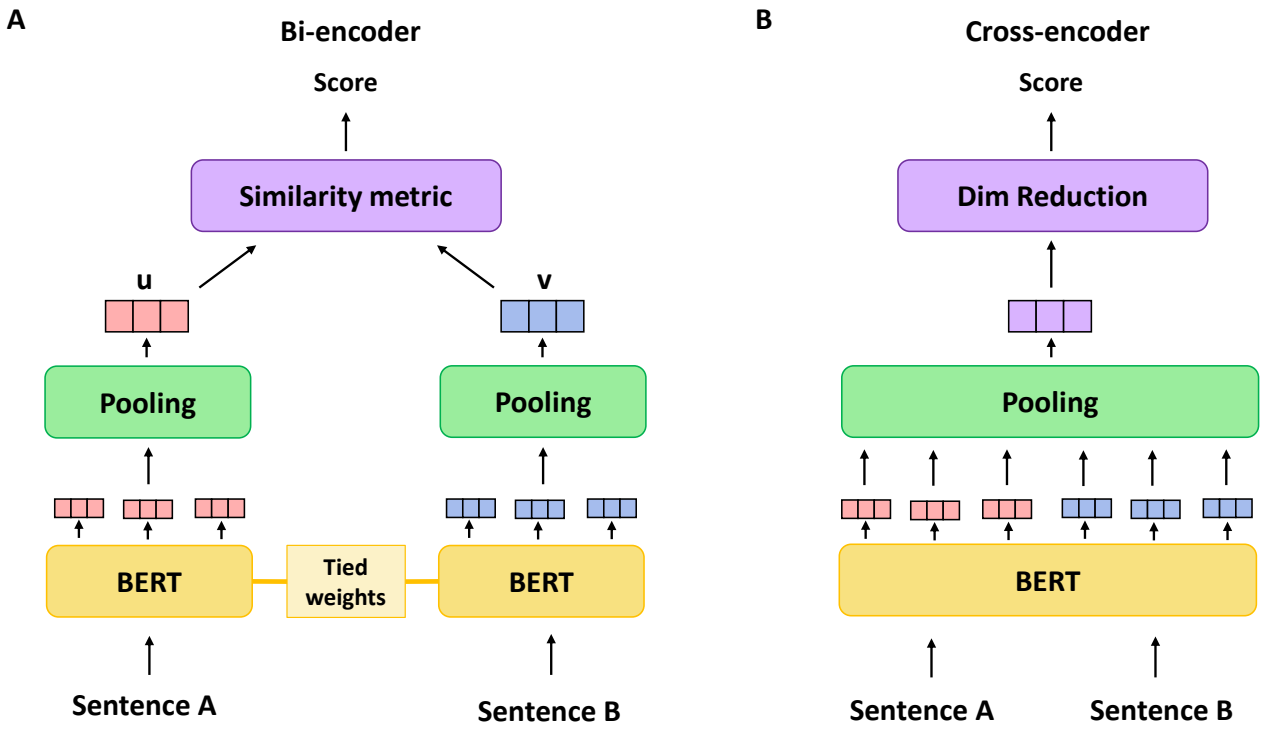


Figure 3.3: Comparison between Bi-encoders and Cross-encoders for extracting sentence embeddings similarity. BERT is the chosen encoder model just for illustration purpose. A) The Bi-encoder encodes the sentences separately. B) The Cross-encoder jointly encodes the sentences in a single transformer, achieving richer interactions between sentences at the cost of slower computation. Adapted from [200].

This independent encoding of the input texts in bi-encoder models allows for several advantages relevant to the objectives of this thesis. First, the dual-encoder design facilitates the independent learning of semantic representations for each input, which also provide the flexibility to compare semantic similarity between diverse length input types, without being constrained by the fixed input size of a single-encoder model.

Another key advantage is the efficient similarity computations enabled by bi-encoder models. The independent encoding of the input texts allows the encoded representations to be pre-computed and stored, enabling fast similarity comparisons at inference time. This efficiency is particularly relevant in the context of the thesis' focus on dimensionality reduction techniques, as it allows for the exploration of compact, low-dimensional representations while maintaining the ability to quickly perform semantic similarity assessments.

Furthermore, the reduced computational complexity of bi-encoder models is a significant

benefit. Compared to cross-encoder Transformer architectures, where the attention mechanism needs to compute a quadratic attention matrix with respect to the input sequence length, bi-encoder models only require two smaller attention matrices, one for each independent input. This reduced computational complexity is particularly beneficial when working with longer input sequences, as it helps mitigate the memory and speed limitations associated with the quadratic attention mechanism in cross-encoder models.

These features of bi-encoder architectures, including their efficiency, computational advantages, and suitability for semantic similarity tasks, make them a promising approach for addressing the limitations and challenges of Transformers, as discussed in Section 2.2.5.

3.3.4 Model Selection

The research incorporates several pre-trained multilingual transformer models to address case studies related to the thesis. Each model has unique strengths that contribute to fulfilling the objectives and research questions in combating false information through advanced computational methods, which are central to the thesis.

- **bert-base-uncased** [133]: Encoder-only model BERT is highly popular and widely applied, making it an excellent candidate to assess response to adversarial attacks and word camouflage. It uses the WordPiece tokenizer, which is adept at minimizing out-of-vocabulary words by breaking them into recognizable subwords.The WordPiece tokenizer, though effective in general, may struggle with unconventional camouflage patterns, which is a critical aspect of this study's focus on robustness.
- **bert-base-multilingual-cased (mBERT)**: BERT [133] transformer model pre-trained on a large corpus of 104 languages Wikipedia articles using the self-supervised masked language modelling (MLM) objective with ∼177M parameters.
- **distilbert-base-multilingual-cased**: Distilled version of the previous model, being on average twice as fast as this model, totalizing ∼134M parameters [201].
- **Language-agnostic BERT Sentence Embedding (LaBSE)**: Language-agnostic BERT Sentence Embedding [202] model trained for encoding and reducing the cosine distance between translation pairs with a siamese architecture based on BERT, a task related to semantic similarity. It was trained over 6 Billion translation pairs for 109 languages. The authors also reported that it has zero-shot capabilities as was able to produce decent results for other not seen languages.
- **paraphrase-multilingual-mpnet-base-v2 (MPNET-base)**: Distilled version of the MPNet model from Microsoft [203] fine-tuned with large-scale paraphrase data using XLM-RoBERTa as the student model. This model is included as a baseline to corroborate the usefulness of the multilingual pre-train in semantic similarity showed in our previous work [204], as it does not includes any fine-tuning on semantic similarity.
- **mstsb-paraphrase-multilingual-mpnet-base-v2 (MPNET-ideal)**: Previous model fitted with multilingual train data from the Semantic Textual Similairty Benchmark (STSb) [205] extended version to 15 languages (mSTSb) [204]. This model has shown

to enhance the performance across languages, outperform monolingual models and the capability of generalize to new tasks. This model has been presented previously in the 22nd International Conference on Intelligent Data Engineering and Automated Learning (IDEAL) [204].

- **bloomz-560m (BLOOMz)** [206]: Fine-tuned variant of the pre-trained multilingual BLOOM [207] and mT5 [208] model families on cross-lingual task mixture of 13 training tasks in 46 languages with English prompts capable of following human instructions in dozens of languages zero-shot. The version used is the version of 560M parameters.
- **xlm-roberta (XLM-R)** [209]: Base-sized XLM-RoBERTa model totalizing ∼125M parameters and large-sized with ∼355M parameters.. XLM-RoBERTa is RoBERTa model [170], robust version of BERT, pre-trained on CommonCrawl data containing 100 languages.
- **pythia-410m-deduped** [210]: Pythia has a decoder-only architecture and employs the BPE (Byte-Pair Encoding) tokenizer. It is selected for its ability to handle a diverse range of English texts from the Pile dataset, suitable for analyzing resilience to Internet language that may include offensive or camouflaged text.
- **mBART-large-50** [211]: Encoder-decoder model mBART's multilingual capabilities and "Multilingual Denoising pre-training", which introduces noise to the input text, may confer resilience against adversarial attacks. This feature makes it potentially more resilient to the presence of rare or unseen characters often used in adversarial camouflage. Uses the SentencePiece tokenizer, incorporating the BPE method, allowing it to process raw text directly, which is beneficial for texts without whitespace word boundaries and for texts with varied formatting.

3.3.5 Fine-tuning

As stated in the Bagkground and State of the art chapter 2, this Thesis follows the well-established "pre-train and fine-tune" paradigm. Instead of training models from scratch, which can be computationally expensive [132], [163], we leverage pre-trained versions of models and further fine-tune them for specific tasks. This approach allows us to benefit from the knowledge captured in the pre-trained models and efficiently adapt them to the objectives of this thesis, particularly in the context of the multilingual fight against misinformation and disinformation.

The fine-tuning process involves multiple iterations of forward and backward passes through the model, where the model's weights are updated in each iteration based on the gradient of the loss function. This iterative process continues until the model performance on a validation set stops improving or until a predetermined number of epochs is reached.

Depending on the task, the Transformer models use different loss functions. For semantic similarity tasks, the mean squared error (MSE) loss between the model's predicted similarity scores and the actual similarity scores (gold labels from the mSTSb dataset) is computed. The objective during fine-tuning is to minimize this MSE loss, thereby adjusting the model parameters to improve prediction accuracy. For classification and named entity recognition

tasks, we use cross-entropy loss, where the models have to optimize parameters to maximize the prediction of a token belonging to one of the possible name entity categories.

During the fine-tuning process, the goal is to obtain the optimal parameter (weight) values from the Transformer architecture that best perform on the specific tasks. However, these parameter values are also influenced by the choice of hyperparameters.

Hyperparameters Optimization

Various hyperparameters are optimized during the fine-tuning process, including the number of epochs, learning rate, batch size, and scheduler settings. This optimization ensures that the model learns effectively from the training data without overfitting or underfitting.

When fine-tuning the pre-trained Transformer models for the downstream tasks in this thesis, we opted for a Bayesian optimization approach instead of relying on traditional grid search or random search techniques. Bayesian optimization is a principled, efficient, and data-driven method for hyperparameter tuning that can outperform these conventional approaches, especially when the hyperparameter search space is complex and high-dimensional [212]. It is based on Bayes' theorem to fit a Gaussian Process model that tries to predict the performance (i.e., the loss) of different hyperparameter configurations, and this prediction is used to inform the selection of future hyperparameters. Additionally, we combined the Bayesian optimization with an Early Stopping technique using the Asynchronous Hyperband algorithm [213], which further enhances the efficiency of the hyperparameter search process.

The key hyperparameters that were explored during the Bayesian optimization experiments, following recommendations and widespread common settings used in the literature and research community [214], [215], include:

- Learning Rate: The learning rate is a crucial hyperparameter that controls the step size taken during the optimization process. We explored a range of learning rates to find the optimal value that would allow the models to converge efficiently without experiencing instability or divergence. The search range for the learning rate was [1e-6, 1e-3].
- Batch Size: The batch size determines the number of samples processed before the model weights are updated. Larger batch sizes can lead to more stable gradients, while smaller batch sizes can introduce more stochasticity and help the model escape local minima. We explored batch sizes in the range of [8, 16, 32, 64]. We also include gradient accumulation of a maximum of 3, which compute gradients for multiple mini-batches before performing a weight update instead of updating the model's weights after each mini-batch
- Number of Epochs: The number of epochs refers to the number of complete passes through the training dataset. We searched for the optimal number of epochs within the range of [3, 10].
- Optimizer: For this thesis, we use of the recommended AdamW optimizer [216] to train downstream the Transformers models [214], [217].
- Scheduler: The learning rate scheduler is responsible for adjusting the learning rate

learning rate	range(1e-6, 1e-3)
scheduler	warmup_steps max 20%
	constant schedule
	constant schedule with warmup
	linear schedule with warmup
	cosine schedule with warmup
	cosine with hard restarts with warmup
	polynomial decay with warmup
epochs	range (1, 10)
	patience = 1600
batch_size	[8, 16, 32, 64]
accumulate_gradient	[0, 3, 5]
optimizer	AdamW
	beta = 0.9
	beta2 = 0.999
	eps = 1e-8
	grad_clip = 1
	l2 wieght_decay = range(0, 1)
dropout	0.1

Table 3.2: Overview of hyperparameters considered during the fine-tuning of models

	PCA	**KPCA**	**ICA**	**Variance Threshold**	**UMAP**
Preprocessor	Standard	Standard		MinMax	MinMax
Scalation	✗	✗	✗	✗	✗
Normalization	✗	✗			
Unsupervised	✗	✗	✗	✗	✗
Feature Selection				✗	
Feature Extraction	✗	✗	✗		✗
Linear	✗		✗		
Non Linear		✗			✗

Table 3.3: Considerations about the previous scaling steps and the characteristics of the different dimensionality reduction techniques applied in this project.

during training, which can help the model converge more efficiently. We explored different scheduler options, including constant, linear, and cosine schedulers, with various warmup strategies and step sizes. The scheduler options explored include a variety of standard schedules with different warmup strategies, such as constant, linear, cosine, and polynomial decay. The warmup ratio was explored up to a maximum of 20% of the total training steps.

- Regularization: To prevent overfitting, we considered regularization techniques, such as L2 regularization and dropout. The L2 weight decay was searched within the range of 0 to 1. The dropout rate was fixed at 0.1, as this value is commonly used and effective in Transformer-based models.

3.4 Datasets

- To study dimensionality reduction on multilingual models (Section 3.1.4), we leveraged the Multilingual Extended Semantic Textual Similarity Benchmark (mSTSb) dataset.
  - The original **Semantic Textual Similarity Benchmark (STSB)** [218] is a dataset designed to evaluate the ability of models to determine the similarity between pairs of sentences, providing a gold standard for semantic similarity tasks in English. It includes pairs of sentences annotated with similarity scores ranging from 0 to 5, reflecting the degree to which the sentences convey the same meaning.
  - We extended the STSB to create the mSTSb[7] [219] by incorporating 16 languages,

[7] https://github.com/Huertas97/Multilingual-STSB

ensuring broad linguistic coverage and enhancing the dataset's applicability to multilingual tasks. This extension involved translating and adapting the original sentence pairs to each of the languages, creating 31 mono- and cross-lingual tasks. Each task comprises pairs of sentences where the model must compute similarity scores, thus testing its ability to understand and process multiple languages simultaneously. The mSTSb train set includes over 5,479 pairs of sentences per task, which we used to fine-tune pre-trained multilingual transformers and fit various dimensionality reduction techniques. The test set, with 1,379 pairs of sentences per task, was employed to evaluate model performance and the impact of dimensionality reduction.

- Related to the case of study of content evasion in Section 3.1.5, multilingual datasets with parallel data, i.e., the same text in different languages, are used to generate content avoidance cases and to evaluate in an unbiased way the performance of the detection models for detecting disguised entities in texts for content avoidance. The use of parallel datasets facilitates a more accurate evaluation of the performance of NER models across all languages, facilitating that the observed model performance across languages is due to their capabilities and not to the existing inequality between languages. The datasets used for this purpose are:
  - **OPUS News-Commentary** [220]: A parallel corpus of political and economic news commentaries in 12 languages was crawled from the web site Project Syndicate provided by WMT.
  - **OPUS ParaCrawl** [220]: Multilingual parallel corpora from around 150k website domains and across 23 EU languages collected in the ParaCrawl project [221] cofinanced by the European Union.
  - **TED2020** [103]: This dataset contains a crawl of nearly 4000 TED and TED-X transcripts from July 2020. The transcripts have been translated by a global community of volunteers into more than 100 languages.
  - **WikiMatrix** [222]: Mined parallel sentences from the content of Wikipedia articles in 85 languages. In this project, a 1.04 score threshold was used for parallel text extraction.
- To investigate the effectiveness of content avoidance techniques on Transformer models in real-world applications related to Section 3.1.6, this thesis focuses on relevant scenarios such as detecting hateful textual content on platforms. This task is a crucial and practical way to test the impact of content evasion and camouflage, while also being at the forefront of efforts to create a healthier online social platform environment.
  - **OffensEval** [223]: English dataset part of the SemEval suite, comprising over 14,000 English tweets that focus on offensive language in social media. The dataset was prepared by removing duplicates, filtering out instances with fewer than three characters, preserving the original text case, and ensuring a balance of binary classes across the dataset to prevent bias.

3.5 Simulating Word Camouflage

This section discusses the methodologies used to study content evasion and the impact of latent space on Transformers, with a focus on word camouflage techniques. The study utilizes a range of techniques to generate camouflaged text for the study of content evasion and impact of latent space of Transformers, mainly through the developed Python package named "pyleetspeak"[8] [224]. This package is designed to handle multiple languages, enhancing its utility for global applications.

The methods considered to simulate word camouflage are referenced in literature [225]–[233] and can be observed in real-world applications on social media[9]:

- **Leetspeak**: This method replaces standard alphabetic characters with numerals or special characters that visually resemble the original letters. For example, replacing "e" with "3" or "a" with "@". This technique alters words to remain recognizable to humans but might confuse text-processing models.
- **Punctuation Insertion**: This involves embedding punctuation marks within words or between letters to disrupt the standard tokenization process without significantly altering human readability. For example, changing "fake" to "f.a.k.e".
- **Syllable Inversion** and Reformatting: Rearranging the syllables of words or adding unnecessary hyphens, spaces, or other characters to disrupt how models interpret the words without losing the word's recognizability to human readers.

The pyleetspeak tool can generate and simulate text for content evasion with annotation as metadata in various formats (JSON, BILOU, IOB), allowing for training Named Entity Recognition (NER) models or other filters. These annotations provide the scientific community with the ability to develop new filters or apply the tool to any dataset. The changes applied are also saved and returned as metadata, enhancing interpretability and data quality control by ensuring the modifications adhere to the intended camouflaging strategies. Additionally, the flexibility of "pyleetspeak" allows for customization of the camouflaging process to meet specific research needs, making it applicable to various text analysis tasks. Furthermore, it is useful as a data augmentation tool, being able to applied the camouflage as a way to generate new version or hardest version of data instances to avoid overfitting during training and better grasping the nuances of the real task objective.

These camouflage techniques are essential for exploring the robustness of Transformer models against content evasion. As these techniques become more popular [234]–[237], it is crucial to explore various scenarios to understand their impact on model performance. The study systematically varies adversarial camouflage attacks across three parameters: complexity level, word camouflage ratio, and instance camouflage ratio, to thoroughly assess the resilience of NLP models against a wide range of adversarial conditions.

- **Complexity level** parameter refers to the degree of modification applied to the words in the dataset, with increasing complexity designed to challenge the model's ability

[8] https://pypi.org/project/pyleetspeak/
[9] Real Social Media examples of camouflage

to process and understand the text correctly. As previously mentioned, complexity is categorized into three levels[10]:

  - Level 1 (Low Complexity): Simple substitutions that slightly alter the appearance of words.
  - Level 2 (Medium Complexity): More extensive use of substitutions and punctuation insertions.
  - Level 3 (High Complexity): Highly intricate modifications that substantially alter the word's appearance.

- **Word camouflage ratio** measures the percentage of camouflaged words in a text instance. For example, a 50% word camouflage ratio means that half of the words in each instance are modified. Varying the word camouflage ratio tests the model's threshold for handling corrupted text within a single instance and helps understand how much alteration a model can handle before its performance degrades significantly.
- **Instance camouflage ratio** parameter indicates the proportion of camouflaged text instances within the entire dataset. For instance, a 25% instance camouflage ratio means that 25% of the dataset's text instances are modified according to the specified complexity level and word camouflage ratio. Adjusting this ratio provides a nuanced view of how NLP models perform under different densities of adversarial inputs, reflecting real-world conditions where varying levels of camouflaging might occur.

  This way, the study aims to provide a comprehensive evaluation of the impact of word camouflage on Transformer models, contributing to the development of more robust NLP systems. This aligns with the thesis's motivation to leverage computational techniques to enhance the resilience and effectiveness of models in detecting and combating false information.

3.6 Dimensionality Reduction Techniques

To explore the impact and potential of dimensionality reduction in the context of multilingual STS tasks, various techniques are employed for reasons related to computational efficiency and semantic understanding:

- **Principal Component Analysis (PCA)** and its variant Incremental PCA (IPCA) are linear feature extraction methods that identifies the principal components capturing the most variance in the data. IPCA is as a variant that is more memory-efficient, making it suitable for large datasets.
- **Independent Component Analysis (ICA)** is introduced as an unsupervised method that seeks components that are maximally independent and non-Gaussian, emphasizing its probabilistic approach to feature extraction.
- **Kernel Principal Components Analysis (KPCA)**, a kernel-based learning method

[10] Examples of Camouflage Level Complexity

for PCA that works by mapping the data into a nonlinear feature space and then performing linear PCA on these patterns. Various kernels like Polynomial, Gaussian RBF, Hyperbolic Tangent (Sigmoid), and Cosine are considered.

- **Variance Threshold**, an unsupervised feature selection method that removes features with variance below a specified threshold, aiming to reduce dimensionality by discarding less informative features.
- **Uniform Manifold Approximation and Projection (UMAP)**. UMAP is based on manifold learning and topological data analysis techniques for unsupervised dimension reduction, focusing on preserving both local and global data structures in lower-dimensional spaces.

Before applying dimensionality reduction techniques such as PCA, KPCA, ICA, Variance Threshold, and UMAP, it is necessary to consider the distribution of features. PCA and KPCA require a Gaussian distribution and normalization to make variances comparable. ICA assumes a non-Gaussian distribution and does not require any preprocessing. For Variance Threshold, using MinMaxScaler is the best standardization method, as it allows the variance selection threshold to affect all dimensions equally. UMAP does not assume a Gaussian distribution but benefits from scaling the features to a given range, which is why MinMaxScaler is applied before UMAP. A summary of the considerations and characteristics of each technique is listed in Table 3.3.

3.7 Evaluation Approaches

3.7.1 Dimensionality Reduction in Latent Spaces

To evaluate the impact of dimensionality reduction techniques on multilingual transformer embeddings in the context of semantic textual similarity (STS), we relied on the multilingual extended STS Benchmark (mSTSb). This benchmark, widely recognized in the field, served as our evaluation methodology. It allowed us to assess the cosine similarity and Spearman correlation coefficient between model-generated similarity scores and gold standard scores, as detailed in subsection 3.7.4. We employed four distinct approaches to thoroughly evaluate the effects of dimensionality reduction on multilingual transformer models.

Approach 1: Pre-trained Models Without Dimensionality Reduction

This approach's objective is to serve as a baseline to understand the performance of models in their pre-trained state without any dimensionality reduction. This approach directly evaluates the pre-trained models on the mSTSb test split to generate embeddings and compute semantic similarity scores. This approach provides a benchmark to compare the effects of dimensionality reduction techniques applied in Approach 3.

Approach 2: Fine-tuned Models Without Dimensionality Reduction

The second approach establishes a baseline for fine-tuned models to assess improvements from task-specific training without dimensionality reduction. Here, pre-trained models are fine-tuned using the mSTSb train split and then evaluated on the mSTSb test split to compute

semantic similarity scores. This approach serves as a benchmark to evaluate the impact of dimensionality reduction on fine-tuned models, as explored in Approach 4.

Approach 3: Reduced Embeddings from Pre-trained Models

The third approach was designed to assess the impact of dimensionality reduction techniques on embeddings generated by pre-trained models. In this formulation, we subjected the embeddings generated by pre-trained models from Approach 1 to various dimensionality reduction techniques. We then evaluated these reduced embeddings on the mSTSb test split. This approach was instrumental in helping us understand how dimensionality reduction can improve or preserve the performance of pre-trained embeddings while reducing computational overhead.

Approach 4: Reduced Embeddings from Fine-tuned Models

The objective of the fourth approach is to evaluate the effect of dimensionality reduction on embeddings generated by fine-tuned models. Embeddings generated by fine-tuned models from Approach 2 are reduced using dimensionality reduction techniques and evaluated on the mSTSb test split. This approach examines whether fine-tuning combined with dimensionality reduction can enhance model performance and efficiency.

These approaches provide a structured framework to compare and analyze the impact of dimensionality reduction on both pre-trained and fine-tuned multilingual transformer models, ensuring a comprehensive evaluation aligned with the research motivation and objectives of the thesis.

3.7.2 Name Entity Recognition of Camouflage for content evasion

To evaluate the effectiveness of multilingual models in detecting content evasion using word camouflage techniques, we approach the problem as a Named Entity Recognition (NER) task. The evaluation process includes selecting representative multilingual data, creating camouflaged versions of the data to simulate evasion methods, developing and comparing multilingual models, benchmarking them against monolingual counterparts, and externally validating the results.

We utilized diverse datasets from different sources, including News-Commentary, ParaCrawl, TED2020, and WikiMatrix (see more details in Section 3.4).

Generating Camouflaged NER Data

New versions of the datasets were created with word entities camouflaged to mimic methods used to evade content moderation systems on social media platforms. The "pyleetspeak" Python package previously explained (see Section 3.5) was used for this purpose. The evaluation focused on the Named Entity Recognition (NER) task, with camouflaged entities annotated using four types: LEETSPEAK, PUNCT_CAMO, INV_CAMO, and MIX (a combination of leetspeak and punctuation camouflage). Leetspeak was applied with a probability of 45%, punctuation camouflage with 25%, inversion camouflage with 10%, and a combination of leetspeak and punctuation camouflage (MIX) with 30%.

Models Training and Comparison

The models were then fine-tuned for the NER task using the Spacy interface, with specific parameters and hyperparameters, to ensure consistency and reproducibility. As detailed in Section 3.7.4, the evaluation metrics included F1-macro, F1-micro, and F1-weighted to account for the imbalanced representation of different NER entities in the dataset.

The evaluation involved comparing the performance of different multilingual transformer models to assess the benefits of semantic-awareness. This included models pre-trained on the Semantic Textual Similarity Benchmark (STS) and those without such pre-training. This comparison aimed to determine whether semantic-aware models perform better in detecting content evasion techniques than non-semantic-aware models. The best-performing multilingual models were compared against monolingual baseline models for each language. This comparison aimed to determine whether multilingual models trained in diverse languages could effectively detect content evasion in various linguistic contexts compared to models trained explicitly in a single language.

External Validation

To further validate the robustness and effectiveness of the best multilingual model, we used the AugLy library [190] for external validation. Although AugLy primarily focuses on data augmentation for text, it provides methods similar to word camouflage techniques. These methods include replacing letters with similar Unicode or non-Unicode characters, inserting punctuation, changing text font, and inverting text. AugLy generated additional camouflaged data not used during training to test the multilingual models' ability to detect novel unseen camouflage strategies.

3.7.3 Camouflage Adversarial black-box attacks

To study the impact of camouflage adversarial black-box attacks on how Transformer models handle inputs that have been deliberately modified to evade detection while remaining comprehensible to humans, two main phases are pursued, each with distinct objectives and approaches using the OffensEval dataset (for more details in Section 3.4).

In Phase I, the methodology focuses on evaluating the impact of various word camouflage techniques on Transformer models, particularly examining their performance without any adversarial training, referred to as Naive models. Naive models are trained using standard, non-adversarial datasets and serve as a baseline to understand how well models perform under normal circumstances and when exposed to camouflaged inputs without any prior exposure or specific defences against such attacks. This phase involves generating adversarial text samples using the pyleetspeak Python package, where the complexity of these camouflage techniques is systematically varied across three levels to comprehensively assess the models' robustness (see Section 3.5). The evaluation setup includes generating 31 test datasets with varying degrees of camouflage complexity and instance ratios, ensuring a thorough examination of model performance under adversarial conditions. Performance is measured using the F1-macro score to isolate the impact of camouflage techniques, consider the performance in the downstream task and not only in the camouflage instances, and minimize class distribution bias.

A crucial aspect of Phase I is the examination of tokenization relevance. The study investigates how different tokenizers (WordPiece, Byte Pair Encoding (BPE), and SentencePiece) affect the models' vulnerability to camouflaged words. By comparing the overlap of tokens from original and camouflaged words, the study assesses the consistency and robustness of each tokenizer (see more details in Section 3.7.4).

Phase II shifts the focus to enhancing model resilience against word camouflage. This phase investigates both external countermeasures, such as MASK and BLANK filters, and intrinsic adversarial training strategies. The MASK filter replaces identified camouflaged words with a [MASK] token, while the BLANK filter removes them entirely, leaving a placeholder. These countermeasures assume a hypothetical perfect detection system that can discern camouflaged from non-camouflaged text, establishing an upper bound for robustness that such preprocessing could achieve. By comparing these idealized filters' performance with naive models, the study aims to determine the maximum potential benefit of external filtering.

Static training involves introducing camouflaged data once, whereas dynamic training continuously updates the data during training. The dynamic approach is hypothesized to better prepare models for real-world scenarios by exposing them to various camouflage patterns. This strategy ensures that models encounter camouflaged and non-camouflaged forms of the same word across different training epochs, enhancing their adaptability. The methodology also incorporates external validation through the AugLy library to demonstrate the efficacy of dynamic training over static methods.

This evaluation methodology outlines a detailed experimental framework to assess model performance under incremental levels of camouflage complexity. Each model's robustness is measured by comparing its performance on camouflaged datasets to its performance on the original dataset, calculating the percentage decrease in performance. This incremental evaluation approach, starting from tokenization and naive models, progressing through external filters, and culminating in fine-tuning with dynamic and static approaches, provides a comprehensive analysis of the models' resilience to adversarial attacks.

3.7.4 Performance Metrics

This methodological approach emphasizes the paramount importance of aligning the semantic content of language models with human perceptions of meaning, a foundational goal of NLP.

Semantic Similarity: The performance is assessed using the Spearman correlation coefficient to compare the similarity scores obtained from the models with the gold standard scores from the mSTSb dataset. The study aims to understand the impact of dimensionality reduction both on pre-trained and fine-tuned model embeddings, focusing on unsupervised methods due to the nature of the embeddings produced by the models.

- Cosine Similarity: For each sentence pair to compute their semantic similarity score, the cosine similarity between the embeddings of the two sentences is calculated. This metric measures the cosine of the angle between two vectors, serving as an indication of how similar they are. The cosine similarity between the two sentence embeddings u and v is a variant of the inner product of the vectors normalised by the vectors' L2 norms,

as shown in equation 3.1. Where N represents the number of dimension of the sentence embeddings u and v, $\langle u, v \rangle$ is the inner product between the two vectors, and $\|.\|$ is the L2 norm.

$$CosSim(u, v) = \frac{\sum_{i=1}^{N} u_i v_i}{\sqrt{\sum_{i=1}^{N} u_i^2} \sqrt{\sum_{i=1}^{N} v_i^2}} = \frac{\langle u, v \rangle}{\|u\| \ \|v\|} \tag{3.1}$$

- Spearman Correlation Coefficient (ρ): The Spearman correlation coefficient is used to measure the statistical relationship between the model-generated similarity scores and the gold standard similarity scores provided in the mSTSb dataset. This metric assesses how well the relationship between the model predictions and actual similarities can be described using a monotonic function.

In the evaluation of Named Entity Recognition (NER) and classification models, especially within multiclass and imbalanced scenarios, several metrics are crucial for assessing the models' effectiveness:

- F1-score: A harmonic mean of precision and recall, providing a balance between the two when evaluating NER systems. It is calculated as shown in equation 3.2, where TP is true positive, FP is false positive, FN is false negative, precision is the ratio of correctly predicted positive observations to the total predicted positives, and recall is the ratio of correctly predicted positive observations to all observations in the actual class.

$$F1 = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.2}$$

- F1-score Micro: Aggregates the contributions of all classes to compute the average F1-score, making it more sensitive to the performance on the majority class. This makes the Micro variant preferable when the primary concern is the overall effectiveness across all predictions.

$$\text{F1 Micro} = \frac{2 \cdot \text{Total TP}}{2 \cdot \text{Total TP} + \text{Total FP} + \text{Total FN}} \tag{3.3}$$

- F1-score Macro: Computes the F1-score independently for each class and then takes the average. This treats all classes equally, emphasizing the importance of performance on rare classes.

$$\text{F1 Macro} = \frac{\sum_{i=1}^{n} \text{F1}_i}{n} \tag{3.4}$$

- F1-score Weighted: Calculates the F1-score for each class individually but averages them, weighted by the number of true instances for each class. This method helps

account for class imbalance by giving more weight to the majority class.

$$\text{Weighted F1} = \sum_{i=1}^{n} w_i \times \text{F1}_i \tag{3.5}$$

where $w_i = \frac{\text{No. samples in class i}}{Totalnumberofsamples}$

- Matthews Correlation Coefficient (MCC) and Cohen's kappa: Both metrics offer robust evaluations independent of class distribution, making them exceptionally useful in imbalanced scenarios. The MCC provides a comprehensive measure that takes into account true and false positives and negatives, offering a balanced metric that is informative even when class distributions are highly skewed. It is effective for evaluating the quality of binary classifications, robust even with class imbalances. It is defined in 3.6. Measures the agreement of prediction with the actual classification, adjusting for the agreement that happens by chance (see eq. 3.7, where p_o is the probability of agreement on the label assigned, and p_e is the expected agreement when lables are randomly assigned, estimated using a prior for each source of label. The Kappa Statistic adjusts for chance agreement, making it reliable for assessing model performance across multiple classes, irrespective of the distribution.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{3.6}$$

$$\kappa = (p_o - p_e)/(1 - p_e) \tag{3.7}$$

- Confusion Matrix: Provides a detailed breakdown of prediction results and the types of errors made by the model. It's a raw metric that offers in-depth insights into the performance across different classes, helping identify if the model is confusing two classes frequently, which is particularly beneficial for fine-tuning NER models.
- In the third article associated with this thesis (see 3.1.6 and 3.7.3, the performance of Transformer models is evaluated using the F1-macro score. This rigorous measure isolates the impact of word camouflage techniques and minimizes class distribution bias. Unlike other studies [238], [239], that use the Error Finding Rate (EFR) – a metric focused on the rate of detecting malignant cases bypassing the model – our study prioritizes the F1-macro score. This ensures that the evaluation considers both altered and original data instances to maintain the model's efficacy on the primary downstream task.
  - Each model's robustness is measured by comparing its performance on camouflaged datasets with varying levels of complexity, word camouflage ratios, and instance camouflage ratios to its performance on the original dataset. The percentage decrease in performance (PD) is calculated using the formula 3.8:

$$PD = \left(\frac{\text{Initial F1 Score} - \text{Final F1 Score}}{\text{Initial F1 Score}} \right) \times 100 \tag{3.8}$$

– Related to the impact on Transformer models' performance, the impact is also measured specifically in the tokenization process. For this purpose the "overlapping" of tokens before and after subtle camouflage modification is considered. This consists of the original word w is tokenized as $T(w) = T_1, ..., T_n$ with n tokens, and the camouflaged word w_C is tokenized as $T(w_C) = T'_1, ..., T'_m$. The coverage of w is the number of T_i tokens from $T(w)$ overlap with T'_i tokens from $T(w_C)$. Values range from 0 (no token overlap) to 1 (identical tokenization), showing the consistency of the tokenization process under random and keywords camouflage conditions.

3.7.5 Statistical Analysis

To determine whether the use of dimensionality reduction techniques significantly affects the models' performance, a statistical comparison is conducted between the baseline approaches (Approaches 1 and 2) and their respective dimensionality reduced counterparts (Approaches 3 and 4) from Section 3.7.1. This comparison involves computing the average Spearman correlation coefficient across all models for each pair of approaches and conducting a two-tailed paired T-test to test the null hypothesis that there is no difference in the average Spearman correlation coefficients between the baseline and reduced approaches.

Additionally, to better evaluate the magnitude of the impact of camouflage in Transformer tokenization presented in Section 3.7.3, we have incorporated Wilson confidence intervals as error bars alongside the average value of overlapping tokens before and after tokenization.

Wilson confidence intervals are used to estimate the precision of the calculated proportions, offering a more reliable measure compared to standard error bars, especially for proportions close to 0 or 1. These intervals were calculated at a 99% confidence level, providing a robust estimate of the expected variability in the observed proportions. The choice of Wilson intervals is due to their ability to stay within the natural limits of proportions ([0, 1]) and their improved precision for datasets with extreme proportions.

3.8 Reproducibility and Open Science

In the pursuit of maintaining reproducibility standards, this thesis employs several state-of-the-art tools designed to share the computational research process, enhancing the transparency of the research and remarking on the thesis's commitment to open science principles.

- CodeOcean[11]: This cloud-based platform is explicitly utilized for reproducing code associated with the second article. It facilitates the seamless reproduction and scaling of computational environments, ensuring that the research underlying the development of evasion techniques and NER models is easily reproducible and transparent. By enabling the replication of the specific computational settings and code used in the study, Code Ocean enhances the credibility and accessibility of the research findings, aligning with the thesis's commitment to open collaboration.

[11] https://codeocean.com/capsule/6133510/tree

- Weights & Biases (W&B)[12][13]: This powerful tool is employed for the systematic tracking of experimental results and the organized development of models. It has been instrumental in streamlining the research process, ensuring that the research adheres to high methodological rigour and transparency standards, thereby contributing significantly to the thesis's open science objectives.
- HuggingFace Hub[14]: Hugging Face Hub has been a critical resource for loading models and spaces. This platform has enabled the sharing and accessing of pre-trained models and research outputs with the wider community, fostering a collaborative environment. The utilization of Hugging Face Hub not only facilitates access to state-of-the-art NLP models but also encourages the sharing of this thesis's contributions, further embedding the research within the global effort to advance NLP technologies in an open and accessible manner.
- GitHub[15]: Indispensable for version control and code shareability, GitHub has been used throughout this Thesis research. It not only facilitated the efficient management and tracking of code changes but also ensured that the developed code is readily accessible to other researchers.
- Datasets: The datasets developed and utilized in this research have been published in reputable conferences [240] and shared in GitHub[16][17], making them available to the academic community.

3.9 General Public Dissemination Activities

Apart from academic congress and workshop participation, the dissemination of the results to the broad public has been one of the objectives to pursuit following open science and trustworthiness of AI.

To fulfill this objective, several activities has been achieved:

- Spain AI talks: Talks on raising awareness about Artificial Intelligence, trends, and opportunities were given at Extremadura Centro Joven, and Secondary High Schools: IES Bioclimático (Badajoz) (2022) and IES Ágora (Cáceres) (2022). More details can be found on Spain AI Twitter[18] [19] [20].
- #Hilotesis competition: The #HiloTesis 2021 contest is an initiative of the Red de Divulgación y Cultura Científica (RedDivulga) of the Comisión Sectorial de I+D+i de Crue Universidades Españolas. The Polytechnic University of Madrid (UPM) promotes

[12] https://wandb.ai/huertas_97
[13] https://wandb.ai/aida-group/ASOC-LeetSpeakNER-full-XX-MultiNER?nw=nwuserhuertas_97
[14] https://huggingface.co/Huertas97
[15] https://github.com/Huertas97
[16] https://github.com/Huertas97/Multilingual-STSB
[17] https://github.com/Huertas97/XX_NER_WordCamouflage
[18] https://twitter.com/Spain_AI_/status/1522515227803324416
[19] https://twitter.com/Spain_AI_/status/1519328409591164928
[20] https://twitter.com/Spain_AI_/status/1523591333117521920

this initiative to foster the development of communication and scientific dissemination skills among future or recent PhDs. Participants are required to disseminate their doctoral thesis in a thread of no more than 20 tweets on Twitter. More information can be found on Twitter.

- COmputing Science and Technology for smArt Cities (COSTAC) (2022-2023-2024): The COSTAC workshop is an initiative of the PhD Program on Computing Science and Technology for Smart Cities at the Technical University of Madrid (UPM) that provides PhD students of the program a platform to present and discuss their most recent and significant findings and experiences in their PhD Thesis.
- Tech Against Terrorims (TAT) & Global Internet Forum to Counter Terrorism (GIFCT) e-learning webinar (2023): Dissemination of the scientific research regarding content evasion simulation and countermeasures was presented in a webinar focused on how online terrorist actors adapt their tactics to combat counterterrorism and content moderation efforts. Details about this webinar can be found here.

3.10 Ethical Considerations

Ethical considerations are crucial in this thesis, especially given the sensitive nature of misinformation, hate speech, and rumors that are included in the research. Our research aims to combat these issues, not promote them. Our work is conducted with a clear intent to contribute positively to the field of AI and societal well-being.

Our primary goal is to counter the spread of false information and harmful content aiming to contribute to a safer and more informed digital environment. Simulating and generating content evasion techniques is intended to create synthetic data, helping us anticipate and counteract malicious tactics effectively. This ensures our models can be robust without exposing real users to harmful content and .

We unequivocally condemn any misuse of our research or tools. The techniques and models developed in this thesis are intended solely for the purpose of improving content moderation and combating the spread of harmful information. We advocate for the responsible use of AI and emphasize the importance of ethical considerations in the deployment of such technologies.

Chapter 4

Article Collection

4.1 Exploring Dimensionality Reduction Techniques in Multilingual Transformers

This instance is the accepted paper via open access that can be accessed in [241].

Huertas-García, Álvaro, et al. «Exploring Dimensionality Reduction Techniques in Multilingual Transformers». Cognitive Computation, vol. 15, n.o 2, marzo de 2023, pp. 590-612. DOI.org (Crossref), https://doi.org/10.1007/s12559-022-10066-8

Cognitive Computation (2023) 15:590–612
https://doi.org/10.1007/s12559-022-10066-8

Exploring Dimensionality Reduction Techniques in Multilingual Transformers

Álvaro Huertas-García[1] · Alejandro Martín[1] · Javier Huertas-Tato[1] · David Camacho[1]

Received: 20 May 2022 / Accepted: 5 October 2022 / Published online: 29 October 2022

Abstract
In scientific literature and industry, semantic and context-aware Natural Language Processing-based solutions have been gaining importance in recent years. The possibilities and performance shown by these models when dealing with complex Human Language Understanding tasks are unquestionable, from conversational agents to the fight against disinformation in social networks. In addition, considerable attention is also being paid to developing multilingual models to tackle the language bottleneck. An increase in size has accompanied the growing need to provide more complex models implementing all these features without being conservative in the number of dimensions required. This paper aims to provide a comprehensive account of the impact of a wide variety of dimensional reduction techniques on the performance of different state-of-the-art multilingual siamese transformers, including unsupervised dimensional reduction techniques such as linear and nonlinear feature extraction, feature selection, and manifold techniques. In order to evaluate the effects of these techniques, we considered the multilingual extended version of Semantic Textual Similarity Benchmark (mSTSb) and two different baseline approaches, one using the embeddings from the pre-trained version of five models and another using their fine-tuned STS version. The results evidence that it is possible to achieve an average reduction of 91.58% ± 2.59% in the number of dimensions of embeddings from pre-trained models requiring a fitting time 96.68% ± 0.68% faster than the fine-tuning process. Besides, we achieve 54.65% ± 32.20% dimensionality reduction in embeddings from fine-tuned models. The results of this study will significantly contribute to the understanding of how different tuning approaches affect performance on semantic-aware tasks and how dimensional reduction techniques deal with the high-dimensional embeddings computed for the STS task and their potential for other highly demanding NLP tasks.

Keywords Dimensionality reduction · Natural language processing · Semantic textual similarity · Multilingual transformers · Language models

Introduction

Natural Language Processing (NLP) includes various disciplines that provide a system with the ability to process and interpret natural language, just like humans use language as a communication and reasoning tool [1]. Due to the recent increases in computational power, parallelisation, the availability of large data sets, and recent advances in Artificial Intelligence, especially in the Machine Learning research field, NLP has been steadily proliferating and has garnered immense interest [1, 2]. In recent years, transformer-based architectures [3] have become an indispensable staple in the NLP field. Transformer models can capture latent syntactic-semantic information and encode text's meaning as contextual vectors in a high-dimensional space referred to as embeddings [2, 4]. In contrast to previous approaches, such as Statistical Natural Language Processing, using the attention mechanism provided by these architectures allows us to consider many characteristics involved in human language.

The tremendous power of transformer-based models in Natural Language Understanding (NLU) and the new models

✉ Álvaro Huertas-García
alvaro.huertas.garcia@upm.es

✉ David Camacho
david.camacho@upm.es

Alejandro Martín
alejandro.martin@upm.es

Javier Huertas-Tato
javier.huertas.tato@upm.es

[1] Departamento de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, Spain

that are continuously being proposed allows us to improve the state-of-the-art results in varied NLP tasks dramatically, including question answering, sentence classification, and sentence-pair regression like Semantic Textual Similarity (STS) [2, 4–6]. The semantic evaluation performed in this STS task is one of the "levels of language" that determines the possible meanings of a sentence by focusing on the interactions among word-level [7]. Given the many different linguistic features involved, it entails a high degree of complexity. In STS tasks, the systems must compute how similar two sentences are considering all these features, returning a similarity score, usually ranging between 0 and 5 [8].

Regarding sentence-pair regression tasks, to overcome the massive computational overhead caused by the quadratic dependence on the input size of the attention mechanism in transformers models [5, 9], the use of siamese architectures is a very effective method for efficiently deriving semantically meaningful sentence embeddings [5, 9]. This approach is also called dual-encoder, bi-encoder or siamese architectures. As explained by Humeau et al. [9], the training of a siamese architecture consists of two pre-trained transformer-based models with tied weights that can be fine-tuned for a specific task like computing separately semantic embeddings for a pair of sentences and measure their similarity using the extensively used cosine similarity function [10].

Despite the practical features of cosine similarity, such as symmetry and spatial interpretation, this similarity metric has complexity $O(N)$: time and memory grow linearly with the number of dimensions of the vectors compared [11]. Thus, dimensionality is a bottleneck for similarity computation and embedding storage [12]. Moreover, an increasing number of studies using ensemble approaches based on the concatenation of embeddings can be found in the literature [13–15], aiming to improve the results in state-of-the-art tasks but accentuating this issue. Given that the application of dimensionality reduction techniques can mitigate this bottleneck, it requires further exploration.

Although in the history of NLP, the focus has mainly been on proposing architectures for English tasks. Nevertheless, interest in developing multilingual NLP tools has grown recently to achieve diversity for transferring NLP capabilities to low-resource languages for which labelled (and unlabelled) data is scarce [16, 17]. The incorporation of tasks in various languages in the SemEval [18] and CLEF [19] competitions is clear evidence of this, surmounting the language bottleneck, but also the increasing number of newly introduced multilingual models [19], such as the recent BLOOM (BigScience Language Open-science Open-access Multilingual) [20] open-sourced model with 176B parameters generated by BigScience which has marked an inflexion point on the research and development of large language models (LLMs).

The present paper seeks to address how different dimensionality reduction techniques impact the performance of pre-computed embeddings from multilingual transformer-based models trained using a siamese architecture focusing on semantic similarity by employing four different approaches where the reduction techniques are compared with the pre-trained and fine-tuned versions of the models.

This research aims to expand the current knowledge further on the effect of dimensionality reduction techniques in Natural Language Processing [21–24]. The following are the new contributions of our research concerning previous studies:

- A more comprehensive range of unsupervised dimension reduction techniques is included, from linear and nonlinear feature extraction to feature selection and manifold techniques.
- The effect of these techniques on pre-trained and fine-tuned pre-computed embeddings in the Semantic Textual Similarity (STS) task is explored.
- In contrast to previous work that explored reduction techniques in classical static word embeddings, this paper investigates the effect of dimension reduction in state-of-the-art contextual-based transformer models.
- Unlike previous works focused on the English language, this research analyses multilingual models, overcoming the language bottleneck for the applicability of dimension reduction of the embeddings of these models.

The remainder of the article is organised as follows: the "Related Work'' section outlines previous work on dimensionality reduction techniques, their application in Deep Learning, and the importance of multilingual semantics in NLP. The "Methodology'' section describes the approaches followed in this research to evaluate the impact of dimensionality reduction techniques, the multilingual models and the dimensionality reduction techniques applied, and the "Experimental Setup'' section the data and processes used to fit and evaluate these techniques. The experimental results are discussed in the "Results'' section. Finally, the "Conclusion'' section summarises the results of this work and concludes.

Related Work

Dimensionality Reduction Techniques

Dimensionality reduction techniques aim to reduce the dimensionality of the data, removing irrelevant and redundant features while preserving critical information for subsequent applications, such as facilitating their visualisation and understanding or leading to more compact models with better generalisation ability [25–27].

There are different non-mutual exclusive criteria to classify dimensionality reduction techniques. Firstly, according to the reduction approach, these techniques can be classified into feature selection and feature extraction techniques [28]. Feature selection involves selecting a subset of original features useful for building models, effectively removing some of the less relevant features from consideration [26, 29]. On the other hand, feature extraction transforms the original data into another feature space with specific criteria, creating new variables by combining information from the original variables that capture much of the information contained in those original variables [25, 28]. Additionally, feature extraction methods can be further subdivided into linear and nonlinear according to the variable combinations applied [30].

Secondly, according to the information available in the datasets, dimensionality reduction techniques can be classified as supervised and unsupervised [25, 26]. Supervised techniques require each data instance in the dataset to be labelled accordingly to the task, whereas unsupervised techniques are task agnostic approaches and do not require labelled data.

Dimensional Reduction of Embeddings

Broadly, embeddings can be categorised as pre-trained or downstream fine-tuned embeddings [31], and pre-computed or on-the-fly embeddings [32, 33].

The first criterion to categorise embeddings is whether they come from pre-trained models for general tasks or are task-specific. Pre-trained embeddings are widely used as a starting point for downstream applications. A clear example is transformer models' current 'Pre-train and Fine-tune' paradigm [31]. Training these models from scratch is prohibitively expensive in many cases. Alternatively, using self-supervised trained checkpoints of these models and their pre-trained embeddings as a starting point to be later fine-tuned for supervised downstream tasks is widely used. Unlike previous works in the literature that have only focused on reduced pre-trained embeddings [21–23], in this work, we are interested in evaluating the impact of dimensionality reduction on both types of embeddings, pre-trained and downstream fine-tuned embeddings.

Pre-computed embeddings in NLP are widespread, i.e. embeddings that may or may not be adjusted to a task but are generated beforehand and not at each use time. A straightforward application of pre-computed embeddings is the semantic search NLP task for locating relevant information from massive amounts of text data [34]. A semantic search task uses semantic textual similarity to compare an input text against a large set of texts from a database to extract relevant related information-typically, a distance metric such as cosine similarity ranks which content should be extracted to an input query. However, computing the database embeddings each time a query is introduced is infeasible. Alternatively, it is preferable to compute the embeddings once, store them, and use these pre-computed embeddings for subsequent requests [33]. With this in mind, it is essential to note the usefulness of reducing embedding dimensions, which can improve their utility in memory-constrained devices and benefit several real-world applications.

As Camastra and Vinciarelli mention [35], using more features than is strictly necessary leads to several problems, pointing out that one of the main problems was the space needed to store the data. As the amount of available information increases, the compression for storage becomes even more critical [12, 36, 37]. Additionally, for the scope of this work, it cannot be ignored that the application of dimensional reduction techniques for reducing pre-computed embedding dimensions neither improves the runtime nor the memory requirement for running the models. It only diminishes the needed space to store embeddings. Besides, it increases the speed of making computations (i.e. to calculate the cosine similarity between two vectors), which also contributes to decreasing the considerable impact on the energy and carbon footprints generated during the production use of the models when pre-computed and stored embeddings are required [38]. Research has tended to focus on implementing bigger and more complex models rather than analysing methods to adjust the vector space to the desired task while conserving the required dimensions [38]. The additional problem is that the storage of high-dimensional embeddings is challenging when dealing with large volume datasets [33]; besides, researchers have already raised awareness of the storage limitation we are about to face if current technology is adopted and the storage utilisation growth rate persists [12].

In terms of performance, dimensionality reduction techniques can also contribute. The performance of Machine Learning (ML) and, in particular, Deep Learning (DL) models is highly dependent on the choice of data representation (or features) to which they are applied [39]. For that reason, much effort during the deployment of ML and DL solutions is dedicated to obtaining a data representation that can support effective learning. As a result, different fields where dimension reduction techniques are combined with ML and DL complex models can be found in the literature. For example, from their application in time series forecasting [40] or health sciences for cancer classification [41] to Natural Language Processing (NLP) to improve the clustering of text documents [42] or sentiment classification of opinion texts [17].

Recently, the study of Raunak et al. [21, 22] has shed more light on the importance of reducing the size of embeddings produced by ML and DL models in the field of NLP. More specifically, these authors focus on reducing the size of classical GloVe [43] and FastText [44] pre-trained word embeddings using PCA-based post-processing algorithms, achieving similar or even better performance than the original embeddings.

Other works on the potential of reducing pre-computed embeddings dimensions have been carried out [23], exploring the effect of Principal Components Analysis (PCA) [45, 46] and Latent Semantic Analysis (LSA) [24] dimensionality reduction techniques as a post-processing step of pre-trained GloVe word embeddings for text classification tasks. These authors also corroborated the usefulness of PCA for obtaining more accurate results with lower computational costs concluding that the PCA method is more suitable than LSA for dimensionality reduction. In the same way, Shimomoto et al. [47] propose solving topic classification and sentiment analysis by using PCA to transform pre-computed word embeddings of a text into a linear subspace. Their results showed the effectiveness of using the PCA subspace representation for text classification. Other authors have already proved this fact [48], showing that the storage, memory, and computation required by these large embeddings typically results in low efficiency in NLP tasks, pointing out the importance of methods to compress pre-computed and pre-trained GloVe and FastText embeddings.

Additionally, researchers have explored dimensional reduction techniques for visualising semantic embeddings of small text corpora [49]. The authors explored four dimension reduction strategies on pre-computed embeddings based on PCA and t-distributed stochastic neighbour embedding (t-SNE) [50], concluding that both methods preserved a significant amount of semantic information in the full embedding.

Similarly, the use of dimension reduction techniques is likewise interesting in Semantic Similarity [27, 37]. As discussed previously, in the Semantic Similarity task, the linear $O(N)$ complexity of cosine similarity is one of the reasons why this distance metric is widely used in the community and this study. However, the complexity of cosine similarity, although linear, is a limitation when dealing with many dimensions and a large amount of data. Precisely, in tasks where cosine similarity is used as a distance metric, the large amount of data handled and the high number of dimensions of the embeddings generated by the models used represent a constraint in both computational time and storage that leaves the door open to the use of dimension reduction techniques.

To the best of our knowledge, in the literature, dimension reduction research on embeddings has focused on statistical methods, such as Bag of Words and Term Frequency-Inverse Document Frequency (TF-IDF) [27, 37], and classical pre-computed word embeddings, including the popular GloVe or FastText embeddings [21–24, 36, 49]. These classical word embeddings are more complex and powerful than statistical methods. However, they are static, word-level, and contextual independent, and their main limitation is that they do not consider what context the word is being used. Moreover, the varied embedding techniques explored are limited, focusing mainly on PCA. Nevertheless, it should be mentioned that novel dimension reduction approaches based on this classical technique [27] and feature selection methods [37] are being proposed. In fact this proves the relevance and importance of further exploring embedding dimension reduction. Likewise, these studies do not include multilingualism in their analyses, being limited to the English language.

Hence, the presented study follows the research line proposed by different authors [21–24] but takes a step forward, including a broader range of techniques and evaluating the capability of **dimensionality reduction techniques** in both **pre-trained and fine-tuned pre-computed embeddings** from state-of-the-art contextual-based transformer models from the recently claimed **multilingual** point of view.

Siamese and Non-Siamese Architectures

Training in non-siamese transformer architectures, known as cross encoders, requires that two sentences be passed to the transformer as input while a target value is predicted [9]. This approach requires feeding the model with both sentences, which causes massive computational overhead because the attention is quadratic to the sequence length. As previously described in the "Introduction'' section, siamese architectures are the main alternative to cope with this massive computational overhead in transformers models. As Reimers and Gurevych [5] measured, in a V100 GPU, finding the pair with the highest similarity in a collection of $n = 10000$ sentences for a non-siamese model requires about 65 h. On the other hand, the same task with a model trained with a siamese architecture is reduced to $\sim$ 5 s.

Although it should also be noted that models trained in a non-siamese way generally obtain better results [51] since they generate the embeddings considering the interaction of both sentences, the type of architecture only affects the computational time to obtain the embeddings. Consequently, the architecture type does not affect the size of the computed embeddings, which makes dimension reduction techniques equally interesting and valuable for both architectures.

Therefore, it is noteworthy that the present research pays special attention to the dimension reduction of the embeddings of transformers models trained with siamese architecture because, in Semantic Textual Similarity (STS) tasks, they are a widely used solution that obtains results in a reasonable time and at a feasible computational cost. However, these reduction techniques are equally helpful for other types of architecture.

Importance of Multilingual Semantics

Semantics has many applications in a wide range of domains and tasks. Recent developments regarding Information Retrieval tasks [34, 51, 52] have demonstrated the potential of combining semantic-aware models along with traditional

baseline algorithms (e.g. BM25) [53]. Moreover, the use of semantic-aware models has proven to be an excellent approach to counteract informational disorders (i.e. misinformation, disinformation, malinformation, misleading information, or any other kind of information pollution) [54–57] or to build automated fact-checking approaches [58]. Semantic similarity can be applied to organise data according to text properties, formally an unsupervised thematic analysis [59]. Following the same criterion, the semantic similarity measurement between a sentence and each word can be applied to extract the keywords with the highest semantic content from the sentence [60]. All these applications rely on measuring semantic textual similarity (STS), making STS a crucial task in NLP.

The language bottleneck is a significant limitation of these semantic-aware solutions [61]. Language constitutes one of the most significant barriers to be addressed since a model's ability to handle multiple languages is essential for its widespread applications. In recent years, in the field of NLP, attention has been paid to multilingual models to achieve diversity in transferring NLP capabilities to low-resource languages [16, 18, 19]. Therefore, in this paper, special attention has been paid to multilingual transformers models, with the immediate goal of covering embedding dimensionality reduction in the semantic textual similarity task in the world's most widely spoken languages, even though these techniques are relevant for monolingual and multilingual transformers.

Altogether, this work aims to broaden our knowledge of semantic-aware transformer-based models by analysing the impact of different dimensionality reduction techniques on the performance of multilingual siamese transformers on semantic textual similarity multilingual tasks. The results of this study will contribute significantly to understanding how different tuning approaches affect performance on semantic-aware tasks and how dimensional reduction techniques deal with the high-dimensional embeddings computed for the STS task from the recently claimed multilingual point of view.

Methodology

The main goal of this research lies in providing a deep analysis of the use of different dimensionality reduction techniques to reduce the size of the output embedding of different multilingual transformer models and their impact on performance.

The following subsections describe in detail the techniques explored to reduce embeddings, present the multilingual transformer models included in this study and the different approaches applied to quantify the reduction margin and its effect on the performance.

Dimensionality Reduction Techniques

As previously mentioned, dimensionality reduction techniques can be grouped according to two non-mutual exclusive criteria. This paper includes a range of types of dimensionality reduction techniques, including linear and nonlinear feature extraction and feature selection techniques. Nevertheless, since the transformers models included in this study are employed in a siamese architecture to determine the degree of similarity between a pair of sentences, they output a pair of non-concatenative embeddings between which the similarity is estimated using the cosine distance. Hence, for each labelled similarity score, there are two non-concatenative embeddings. For this reason, even though we have labelled the data, only unsupervised methods are explored.

The dimensionality reduction techniques explored in this project are:

- **Principal Component Analysis (PCA)**: Principal Component Analysis [45, 46] is a powerful unsupervised linear feature extraction technique that computes a set of orthogonal directions from the covariance matrix that capture most of the variance in the data [62]. This is, it creates new uncorrelated variables that maximise variance, and at the same time, most existing structure in the data is retained. It is also important to note that this research uses a variant of PCA known as Incremental Principal Components Analysis (IPCA) [63]. This variant follows the same basic principles as PCA. However, it is much more memory efficient, as it applies PCA in batches, avoiding storing entire data in memory and allowing PCA to be applied on large datasets.
- **Independent Component Analysis (ICA)** [64]: Independent Component Analysis is an unsupervised feature extraction probabilistic method for learning a linear transformation to find components that are maximally independent between them and non-Gaussian (non-normal), but at the same time, they jointly maximise mutual information with the original feature space.
- **Kernel Principal Components Analysis (KPCA)** [65]: Kernel-based learning method for PCA. It uses kernel functions to construct a nonlinear version of the PCA linear algorithm by first implicitly mapping the data into a nonlinear feature space and then performing linear PCA on the mapped patterns [62]. The kernels considered in this project are the Polynomial, Gaussian RBF, Hyperbolic Tangent (Sigmoid), and Cosine kernels.
- **Variance Threshold**: Unsupervised feature selection approach that removes all features with a variance below a threshold. Indeed, this technique selects a subset of features with large variances, considered more informative, without considering the desired outputs.

- **Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)**: The authors of UMAP [66] describe it as an algorithm that can be used for unsupervised dimension reduction based on manifold learning techniques and topological data analysis. In short, it first embeds data points in a new nonlinear fuzzy topological representation using neighbour graphs. Secondly, it learns a low-dimensional representation that preserves the complete information of this space, minimising Cross-Entropy. Compared to its counterparts, such as t-SNE, UMAP is fast, scalable, and allows better control of the desired balance between the local and global structure to be preserved. Two main parameters play a vital role in controlling this: (1) the number of sample points that defines a local neighbourhood in the first step, and (2) the minimum distance between embedded points in low-dimensional space to be clustered in the second step. Larger values of the number of neighbours tend to preserve more global information in the manifold as UMAP has to consider more prominent neighbourhoods to embed a point. Likewise, larger minimum distance values prevent UMAP from packing points together and preserving the overall topological structure.

According to the previous preprocessing steps required before dimensionality techniques, it should be noted that PCA and KPCA assume a Gaussian distribution, and the features must be normalised; otherwise, the variances will not be comparable. Therefore, the StandardScaler is applied beforehand. Regarding ICA, non-Gaussian distribution is assumed, and the data is already withened by the algorithm, so no previous preprocessing step is necessary. For the Variance Threshold, the best standardisation method is MinMaxScaler, as it transforms all features to the same scale but does not alter the initial variability. This allows the variance selection threshold set to affect all dimensions equally. Finally, since there are no Gaussian assumptions under UMAP and the cosine distance calculation benefits from scaling the features to a given range, MinMaxScaler is applied before UMAP. A summary of the necessary considerations about the above scaling steps and the characteristics of the dimensionality reduction techniques applied in this project are listed in Table 1.

Finally, it is worth mentioning that these dimensionality reduction techniques and preprocessing algorithms come from *scikit-learn v1.0.2* [67], except for UMAP, which belongs to *umap-learn* v0.5.2 [66]. For the sake of reproducibility, the different parameters and values used in the experiments are presented in Table 2. Finally, the variance threshold filters for the Variance Threshold technique tested are extracted by previously calculating the variance of each feature (i.e. 768 variances for 768 embedding dimensions), extracting the deciles, and including the maximum and minimum of these variances.

Table 1 Considerations about the previous scaling steps and the characteristics of the different dimensionality reduction techniques applied in this project

	PCA	KPCA	ICA	Variance Threshold	UMAP
Preprocessor	Standard	Standard		MinMax	MinMax
Scalation	✗	✗	✗	✗	✗
Normalisation	✗	✗			
Unsupervised	✗	✗	✗	✗	✗
Feature Selection				✗	
Feature Extraction	✗	✗	✗		✗
Linear	✗		✗		
Non Linear		✗			✗

Multilingual Models

Regarding the models included, we have tried to include the open-source architectures widely used by the NLP community, such as the classic BERT and its robust version of multilingual nature, such as XLM-RoBERTa. The effects of dimensionality reduction and fine-tuning process were explored in the following pre-trained multilingual models extracted from Hugging Face [68]:

- **bert-base-multilingual-cased**: BERT [4] transformer model pre-trained on a large corpus of 104 languages Wikipedia articles using the self-supervised masked language modelling (MLM) objective with ∼177M parameters.

Table 2 Parameters with non-default values used in the previous scaling steps and the dimensionality reduction techniques applied in this project

Technique	Parameters
ICA	random_state = 0
	max_iter = 320
	whiten = True
	tol = 5e-4
KPCA	kernels = [sigmoid, polynomial, rbf, cosine]
	eigen_solver = arpack
	copy_X = False
	random_state = 0
Variance Threshold	threshold = [Min, Max, Decile of variance]
UMAP	pre-computed_knn = True
	metric = cosine
	min_dist = 1
	n_neighbors = [5, 10, 50, 100, 125]
	angular_rp_forest = True

- **distilbert-base-multilingual-cased**: Distilled version of the previous model, being on average twice as fast as this model, totalizing ~134M parameters [69].
- **xlm-roberta-base**: Base-sized XLM-RoBERTa [70] model totalizing ~125M parameters. XLM-RoBERTa is RoBERTa model [71], a robust version of BERT, pre-trained on CommonCrawl data containing 100 languages.
- **xlm-roberta-large**: Large-sized XLM-RoBERTa [70] model totalizing ~355M parameters.
- **LaBSE**: Language-agnostic BERT Sentence Embedding [72] model trained for encoding and reducing the cosine distance between translation pairs with a siamese architecture based on BERT, a task related to semantic similarity. It trained over 6 billion translation pairs for 109 languages. The authors also reported that it has zero-shot capabilities, producing decent results for other not seen languages.

Transformer-based models embed textual information into vectors of high dimensionality. The pre-trained multilingual models in this study generate embeddings with 768 dimensions and 1024 in the case of *xlm-roberta-large*. These default dimensions are considered to be reduced since a different number of dimensions from the default would entail losing the knowledge acquired during the self-supervised pre-training phase (e.g. mask word prediction), in which we are interested in studying the effects of the dimension reduction techniques.

Evaluation Approaches

We have followed four approaches to evaluate and quantify the reduction margin and its effect on performance using different methodologies. These four approaches are shown in Fig. 1.

We have included two different baseline approaches, one using the embeddings from the pre-trained version of the five previously mentioned models (i.e. Approach 1) and another using their fine-tuned STS version (i.e. Approach 2). These baselines will allow us to have a standard to examine the dimensionality reduction capability and its effect on STS task performance.

An approach that applies dimensionality reduction to the generated embeddings is used for each baseline. Consequently, Approach 3 applies dimension reduction using Approach 1 as the baseline. Similarly, Approach 4 uses Approach 2 as the baseline.

These four approaches are described as follows:

- **Approach 1 — Pre-trained models**. In the first approach, we employ and directly evaluate the pre-trained models in the mSTSb test split without applying any dimensionality reduction. This approach is used as the baseline for Approach 3.
- **Approach 2 — Fine-tuned models**. In this second approach, the pre-trained models are fine-tuned downstream using the mSTSb train split and evaluated in the mSTSb test split without applying any dimensionality reduction technique. This approach is used as the baseline for Approach 4. The fine-tuning process will be discussed in more detail in the "Transformers Fine-tuning'' section.
- **Approach 3 — Reduced embeddings from pre-trained models**. In this approach, the embeddings generated by the pre-trained models from Approach 1 in the mSTSb train split are used to fit the different dimension reduction techniques and evaluate them in the mSTSb test split. Thus, an analysis between the results achieved in

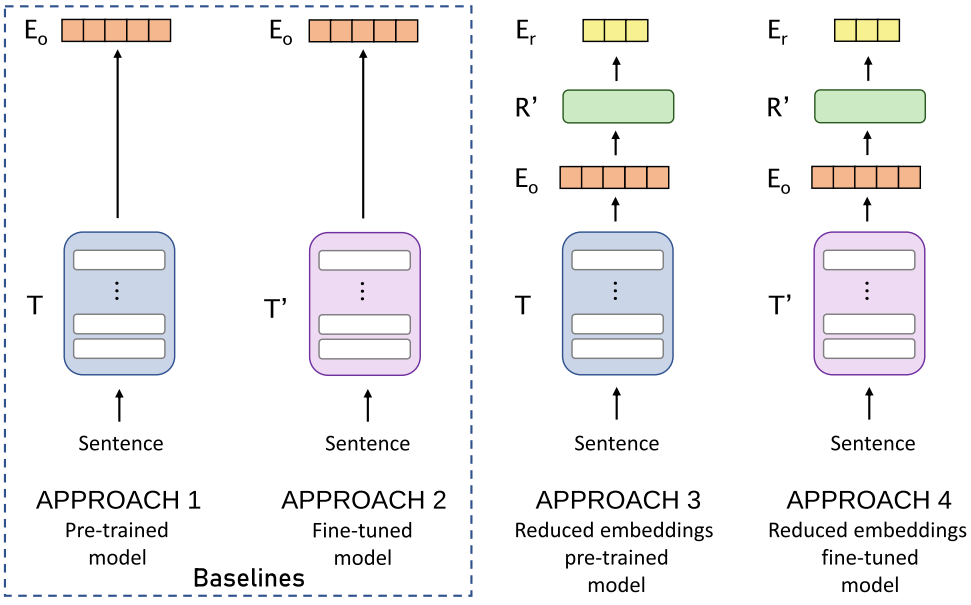


Fig. 1 Representation of the approaches followed to evaluate the impact of different dimensionality reduction techniques in multilingual transformers. Where *T* represents the pre-trained transformer model, *T'* the fine-tuned transformer model, *R'* the fitted dimensionality reduction technique, E_o the original output embedding, and E_r the reduced embedding

Approach 1 and Approach 3 will help to understand the impact of dimensionality reduction techniques in the embeddings from pre-trained models.

- **Approach 4 — Reduced embeddings from fine-tuned models**. This approach is equivalent to Approach 3 but uses the fine-tuned models in Approach 2, allowing us to assess the impact of dimensionality reduction techniques in fine-tuned embeddings.

The experimental design of these approaches will be discussed in more detail in the "Baseline Approaches: Approach 1 and Approach 2'' and "Dimensionality Reduced Techniques Fitting: Approach 3 and Approach 4'' sections.

Experimental Setup

Data

The multilingual extended STS Benchmark (mSTSb) [51] train set is used for fine-tuning the multilingual transformers and fitting the variety of dimensional reduction techniques. This split comprises 16 languages[1] combined in 31 mono- and cross-lingual tasks with 5, 479 pairs of sentences each one. Likewise, the mSTSb test set is used to evaluate the performance of the models obtained from the different approaches. The mSTSb test set comprises 31 multilingual tasks with 1, 379 pairs of sentences per task.

To evaluate the performance in mSTSb, the sentence embeddings for each pair of sentences are computed and the semantic similarity is measured using the cosine similarity metric. Then, the Spearman correlation coefficient (ρ or r_s) is computed between the scores obtained and the gold standard scores, as it is recognised as an official metric used for semantic textual similarity tasks [73, 74].

It is important to note that the mSTSb data variety available for fitting (i.e. train split) totals +183 k sentences (i.e. 16 languages with 5, 749 pairs of sentences each). For linear PCA, this dataset is too large to fit in memory. To manage this situation, an Incremental PCA (IPCA) approach [63] is applied. As previously mentioned, IPCA simply fits the PCA in batches, independent of the number of input data samples but still dependent on the input data features.

Similarly, KPCA and UMAP are computationally more expensive than their DIFaddend linear counterparts [62, 65]. For this reason, these dimensionality reduction techniques were fitted using a subset of $10k$ pairs of sentences (i.e. $20k$ sentences), always ensuring the number of data instances is larger than the number of dimensions. To perform this subsampling, the following requirements were taken into account: (1) all 16 languages must be equally represented, giving a total of 625 sentence pairs for each language; (2) all sentences present in the original train split will be present at least once in some language; (3) the representation of the different sentence pairs must be as random as possible. Following these criteria, we perform a sampling based on assigning sentences to a randomly selected language until we reach the maximum number of sampled data. The different sentence pairs are shuffled randomly at each iteration to avoid bias in the order in which the sentences are assigned to the languages. As each language reaches the maximum data, that language is discarded. This ensures a random distribution of samples in each language but includes the full range of sentences in the original train data.

Computational Resources

Regarding the computational resources used in this study, an Intel(R) Xeon(R) Bronze 3206R CPU at 1.90GHz is used to fit the reduction techniques in Approaches 3 and 4. On the other hand, the pre-computed embeddings used in the approaches and the fine-tuning experiments from Approach 2 are performed using a Quadro RTX 8000 48GB GPU.

Baseline Approaches: Approach 1 and Approach 2

As shown in Fig. 2, baseline Approaches 1 and 2 share similar experimental designs. Both employ the mSTSb test data split to generate the embeddings of the sentence pairs. As mentioned previously, because they are siamese-trained models, one embedding is computed for each sentence in the sentence pair. Then their similarity value is calculated with the cosine distance, and the Spearman coefficient is computed using the reference values to obtain the baseline performance for each multilingual model. The only difference between Approach 1 and Approach 2 is the version of the model used. In the case of Approach 1, it is the pre-trained version of the model, while in Approach 2 it is the fine-tuned version in the mSTSb task. This fine-tuning process is explained in more detail in the following subsection.

Transformers Fine-tuning

The fine-tuning process of the models based on the siamese training strategy for Approach 2 was carried out following the methodology described by Reimers et al. [5, 61]. As depicted in the fine-tuning process of Approach 2 in Fig. 2, for each sentence pair from the mSTSb training split, two networks with tied weights from the same transformer model compute the embeddings for each sentence separately. Then the cosine similarity between the two sentence embeddings is calculated. Finally, the mean squared error (MSE) loss between the predicted and the gold similarity scores are used as the

[1] ar, cs, de, en, es, fr, hi, it, ja, nl, pl, pt, ru, tr, zh-CN, zh-TW

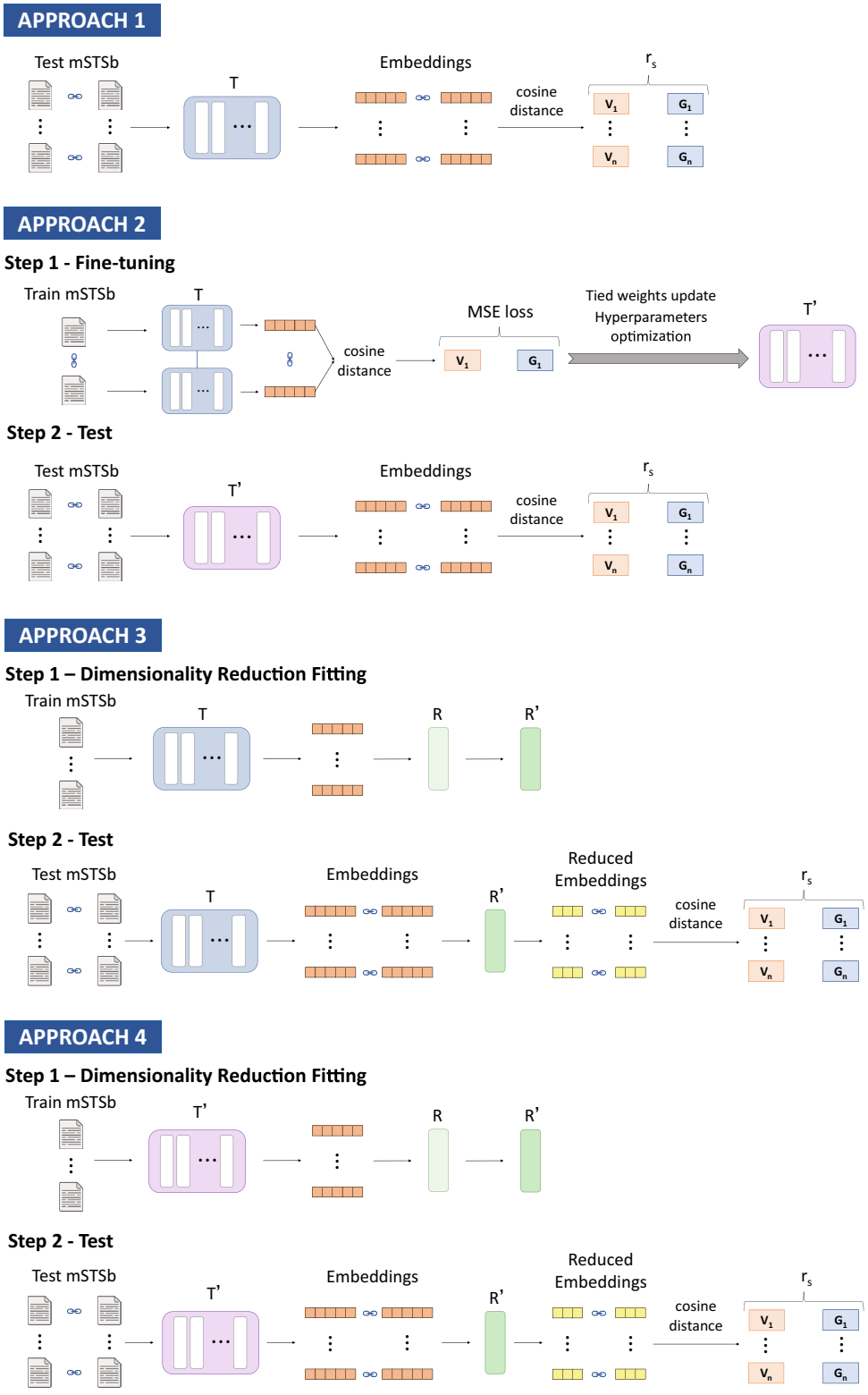


Fig. 2 Diagram of the methodology followed for testing the different Approaches on the multilingual STS benchmark. Where *T* represents the pre-trained transformer model, *T'* the fine-tuned transformer model, *R* the dimensionality reduction technique, *R'* the fitted dimensionality reduction technique, V_i the cosine similarity score computed for *i*th sentence pair, G_i the gold standard similarity score for *i*th sentence pair, MSE for Mean Squared Error loss, and r_s the Spearman correlation coefficient

objective function to update the tied weights. During training, the following hyperparameters were optimised: number of epochs, scheduler, weight decay, batch size, warmup ratio, and learning rate. The hyperparameter values explored, the required time for fine-tuning, and the results of the experiments can be consulted in Table 8 and Weight and Biases [2].

[2] https://wandb.ai/huertas_97/PaperDimRed

Dimensionality Reduced Techniques Fitting: Approach 3 and Approach 4

Figure 2 provides an overview of the process of applying and evaluating dimensional reduction techniques in Approaches 3 and 4. Both approaches follow the same methodology; the only difference is the pre-trained or fine-tuned version of the model used to compute the embeddings for fitting the dimensional reduction technique. The experimental design has two steps, and it is repeated for each number of dimensions explored for each technique and each model. A wide range of the number of dimensions is explored for each dimensionality reduction technique, as shown in the "Results'' section.

Firstly, a dimensionality reduction technique is fitted through the embeddings computed by a multilingual model using the mSTSb train split. As these are unsupervised techniques (i.e. the gold similarity scores are not included), this training step does not consider the relationship between pairs of sentences. However, it uses individual sentences to fit the technique. In other words, if the train split of mSTSb contains $10k$ pairs of sentences from 16 languages, $20k$ separately sentences are used to fit a dimensional reduction technique.

Finally, the embeddings from paired sentences from the mSTSb test split are reduced with the fitted technique. The cosine similarity distance function is applied to obtain a similarity value, and finally, the Spearman correlation coefficient to the gold similarity scores is computed. In contrast to the previous fitting step, in this second step, the relationship between pairs of sentences is considered as we need to evaluate the techniques on mSTSb task.

Statistical Comparison

Additionally, to test if the use of reduced embeddings has a significant impact on the performance in comparison to the baseline approaches, we compare the average Spearman correlation coefficient of the five multilingual siamese transformer models (see the "Multilingual Models'' section) between each pair of baseline and reduced approaches (i.e. Approach 1 vs Approach 3, Approach 2 vs Approach 4). For this purpose, as we are comparing the same set of models in different approaches, the two-tailed paired T-test using a significance level of 0.05 is conducted to test the null hypothesis of identical average Spearman correlation coefficient scores.

Results

This section aims to summarise the effect of a wide variety of dimensionality reduction techniques on the performance of multilingual siamese transformers by comparing the baseline approaches (i.e. Approaches 1 and 2) with the reduced approaches (i.e. Approaches 3 and 4) for each model independently. It must be noted that this work does not pretend to provide a comparative analysis between the different models presented in the "Multilingual Models'' section or to identify the best model for this task. In contrast, this work focuses on applying these dimensionality reduction techniques to reduce the dimensionality of the models' embeddings. Thus, applying different dimensionality reduction techniques does not affect the execution or the memory requirement for running the models. It only diminishes the needed space to store embeddings and increases the speed of computing the cosine similarity between them.

Due to space reasons, average results across the 31 monolingual and cross-lingual tasks are presented instead of a breakdown by language. The average of Spearman correlation coefficients is computed by transforming each correlation coefficient to a Fisher's z value, averaging them, and then back transforming to a correlation coefficient.

Approach 1 vs Approach 3: Dimensionality Reduction in Pre-trained Embeddings

As can be seen in Fig. 3, for every model, the pre-trained performance on mSTSb (i.e. Approach 1) is improved using different dimensional reduction techniques. These results prove that dimension reduction techniques can somehow adjust the knowledge present in the pre-trained embeddings to the semantic similarity task. This fact becomes even more significant in the case of LaBSE, a model with zero-shot capabilities trained on a task close to semantic similarity, which also greatly benefits from the use of dimension reduction techniques, increasing by 0.4 points the Spearman correlation coefficient (see Fig. 3 and Table 3). For the rest of the models, it is equally remarkable that the dimension reduction techniques improve the pre-training performance, almost doubling the score in the models with the XLM-RoBERTa architecture.

Clearly, the best technique in Approach 3 is ICA. Not only because it obtains the most remarkable improvement in pre-training performance for all models, as shown in Table 3, but also because it is the technique that most quickly and with the fewest dimensions overcomes the pre-trained models of Approach 1 (see Table 5). From the table results, the ICA technique improves the pre-trained Approach 1 performances reducing an average of $91.58\% \pm 2.59\%$ of the initial dimensions retaining 100% of the baseline Approach 1 performance. Remarkably, the two-tailed paired T-test comparing Approach 1 vs Approach 3 using the values for this technique from Table 3 resulted in $p = 0.041$, indicating that the performance improvement is significant when using ICA as a dimensional reduction technique. These findings corroborate the ideas of Raunak

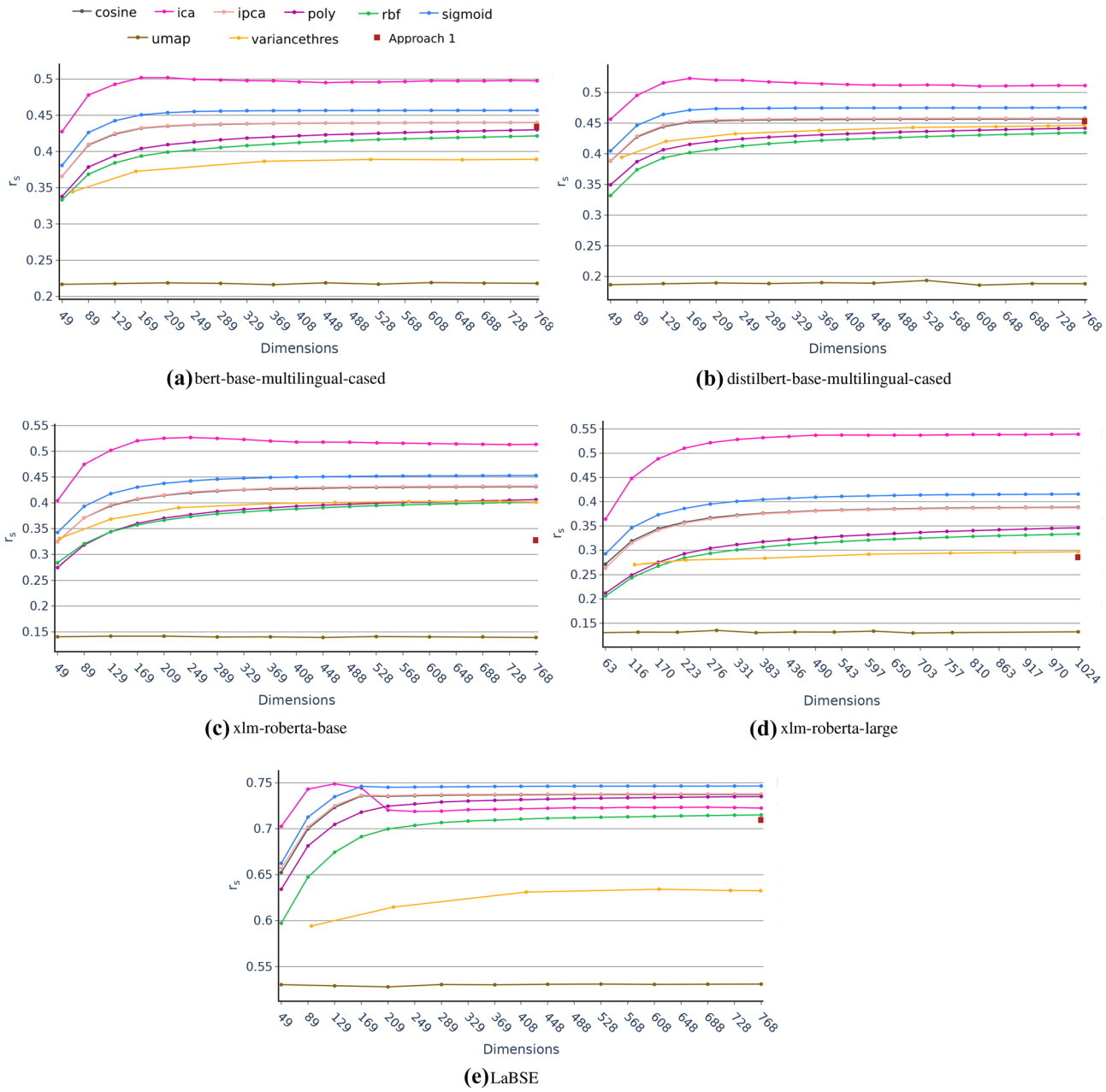


Fig. 3 Approach 3 average Spearman r_s correlation coefficient in multilingual tasks from the mSTSb test as a function of the number of dimension for the different dimensionality reduction techniques grouped by model

et al. [21], who maintained that reduced word embeddings could achieve similar or better performance than original pre-trained embeddings.

The most likely explanation for these results is the difference in the objective of ICA from the other feature extraction techniques. Even though they all transform the initial

Table 3 Average Spearman r_s correlation coefficient comparison between Approach 1 (Ap. 1) and best dimensional reduction technique in Approach 3 (Ap. 3) for the multilingual Transformers

Model	Ap. 1 r_s	Best Technique	Dimensions	Ap. 3 r_s	Fitting Time
bert-base-multilingual-cased	0.4342	ICA	209	0.5019	4 m 16 s
distilbert-base-multilingual-cased	0.4531	ICA	169	0.523	2 m 47 s
xlm-roberta-base	0.3274	ICA	249	0.5269	7 m 51 s
xlm-roberta-large	0.2855	ICA	1024	0.5392	31 m 22 s
LaBSE	0.7096	ICA	129	0.7488	2 m 27 s

space through combinations of dimensions into a new space, the ICA technique is based on optimising mutual information [64, 75]. It tries to find a space where the new features (latent variables) are as independent as possible from each other but as dependent as possible on the initial space. Therefore, in the case of ICA, unlike other techniques such as PCA or KPCA, a higher number of components does not necessarily mean an increase in the information retained or an improvement in the result (as can be seen in Fig. 3 and clearly in Fig. 3 where there is a decrease from 169 dimensions onwards). This would explain why a low number of dimensions would outperform Approach 1.

Likewise, the fact that it is the technique that achieves the best results in Approach 3 in all models could be due to the assumptions and characteristics of both the ICA and the pre-trained embeddings. First, the pre-trained embeddings probably include non-relevant and noisy variables as these embeddings are not adjusted to the STS task. Secondly, since ICA is a technique in which the original variables are related linearly to the latent variables but for which the latent distribution is non-Gaussian, the noise present in pre-trained embeddings agnostic of the STS task could be managed appropriately.

Interestingly, these results also emphasise that the issue of non-Gaussianity is more relevant than the nonlinearity issue. Non-Gaussianity would be more important than how the initial variables are combined, as the ICA technique outperforms linear PCA and nonlinear KPCA. This is in good agreement with other studies comparing the performance of PCA and ICA as a method for feature extraction in visual object recognition tasks [76, 77].

Additionally, the presence of noisy variables in the pre-trained embeddings would also be corroborated by the low scores obtained from the Variance Threshold feature selection technique, which entirely depends on the original variables and cannot manage these noisy distributions.

Consequently, ICA shows excellent properties for obtaining compacted embeddings versions of pre-trained models with a significant decrease of dimensions that improve the result in the task of semantic similarity at a multilingual level.

For all these reasons, we can understand unsupervised dimensionality reduction, specially ICA, as a method of fitting pre-trained models for downstream tasks. As it will be seen in the next section and as might be expected, this unsupervised dimensionality reduction downstream fitting is not comparable to a supervised fitting such as the fine-tuning of Approach 2. However, downstream fitting by unsupervised dimensionality reduction techniques may present interesting advantages such as the fact that being unsupervised is task agnostic resulting in models with higher generalizability and with a lower number of dimensions. Also, these dimensionality reduction techniques do not require GPUs and apply a more interpretable methodology than a Deep Learning model fine-tuning such as transformers.

Approach 2 vs Approach 4: Dimensionality Reduction in Fine-tuned Embeddings

Although it was stated at the beginning of this section that model comparison is not an objective of the paper, different versions of the same architecture have been included for a more comprehensive evaluation of the effects of dimensionality reduction techniques (i.e. *xlm-roberta-base* and *xlm-roberta-large*, *bert-base-multilingual-cased* and *distilbert-base-multilingual-cased*) have been included in the study. Based on the complexity and learning potential of the models, one would expect the *xlm-roberta-large* model to perform better than the *xlm-roberta-base* model. Similarly, the *bert-base-multilingual-cased* model would be expected to be superior to the *distilbert-base-multilingual-cased* model. In Approach 1, however, the opposite is true. Only when fine-tuning occurs in the task (Approaches 2 and 4) is it observed how the performance of the models is in line with the expected complexity (see Table 4). These results provide wider support for the importance of supervised fine-tuning.

Similarly, fine-tuning also alters the impact of dimensionality reduction techniques on the results of multilingual models. Compared to Approach 3, when fine-tuning, the feature selection techniques and nonlinearity become more important, ICA becomes less critical, and the minimum number of dimensions that outperform the baseline approach increases.

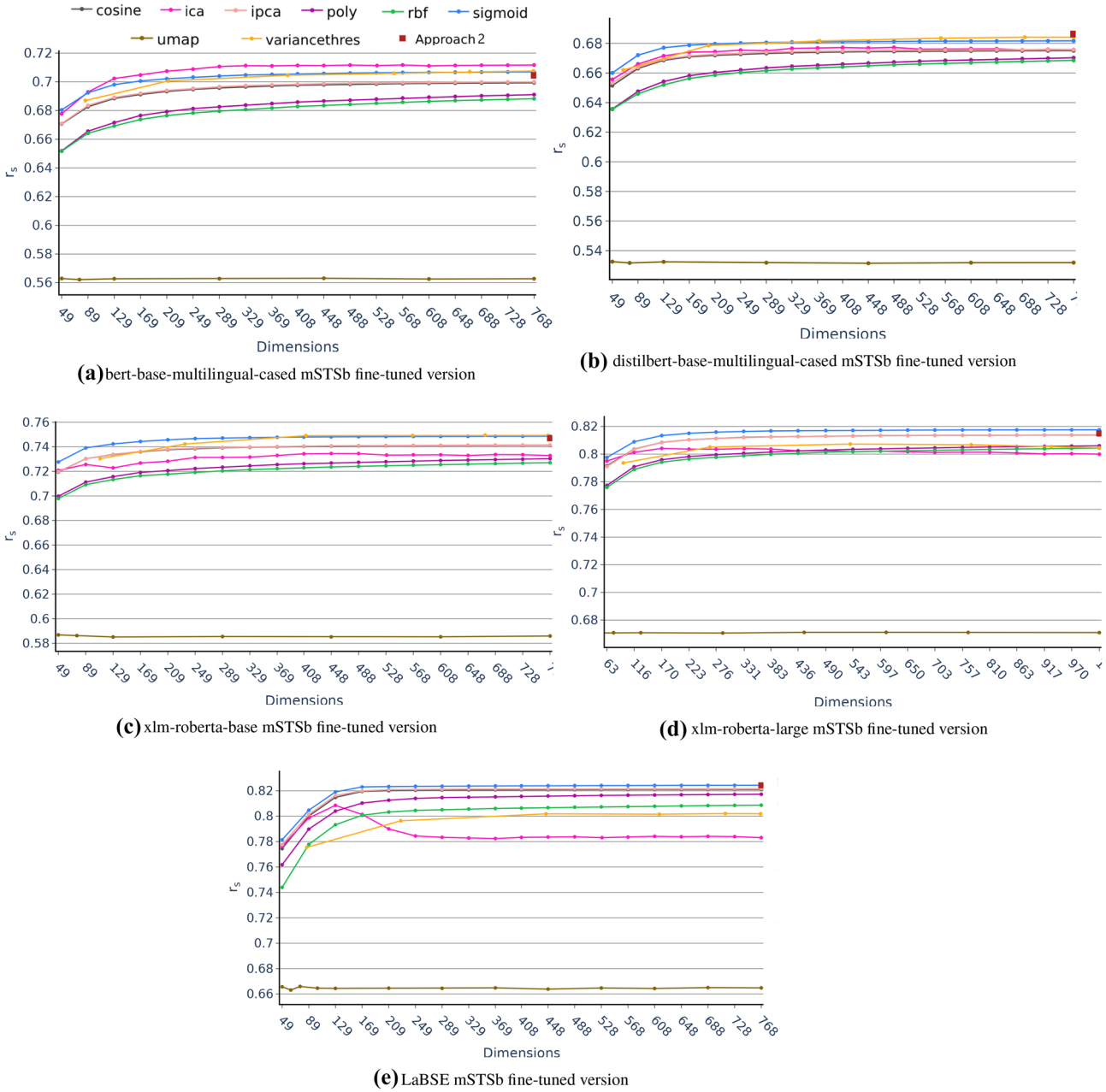
As can be seen in Fig. 4 and Table 4, the promotion of Variance Threshold feature selection as one of the best techniques for some models in Approach 4 could be attributed to the fact that fine-tuning adjust the embeddings to the task, reducing the presence of noisy variable and taking advantage of the variable selection process. This would be in line with the results obtained in Approach 3, where feature extraction techniques more adequately handled the presence of unadjusted variables. This further supports the argument that they can reduce dimensions and generate new feature representations to help improve performance on learning problems.

Furthermore, Table 6 shows the lack of ICA dominance and the emergence of the KPCA-sigmoid technique as the method with fewer dimensions improves the baseline Approach 2 ($59.32\% \pm 29.92\%$ reduced dimensions retaining $99.00\% \pm 2.00\%$ baseline performance). It reveals that managing the non-Gaussianity issue is less relevant than the nonlinearity issue after fine-tuning. The fine-tuning process also impacts the reduction capabilities of the dimensionality reduction techniques since considering the technique that retains the maximum performance with the lowest dimensions for each model in Table 6 shows that the initial dimensions from the baseline Approach 2 are reduced

Table 4 Average Spearman r_s correlation coefficient comparison between Approach 2 (Ap. 2) and best dimensional reduction technique in Approach 4 (Ap. 4) for the multilingual Transformers

Model	Ap. 2 r_s	Best Technique	Dimensions	Ap. 4 r_s	Fitting Time
bert-base-multilingual-cased-fine-tuned	0.7045	ICA	568	0.7117	12 m 38 s
distilbert-base-multilingual-cased-fine-tuned	0.6863	VarThres	692	0.6842	2 s
xlm-roberta-base-fine-tuned	0.7470	VarThres	673	0.7495	3 s
xlm-roberta-large-fine-tuned	0.8150	KPCA-sigmoid	1024	0.8176	20 m 6 s
LaBSE-fine-tuned	0.8242	KPCA-sigmoid	768	0.8243	19 m 25 s

by an average of 54.65% ± 32.20%. Although this average reduction is lower than the achieved earlier in the comparison of Approach 3 with the baseline Approach 1, it is still remarkable that even after fine-tuning, the multilingual performance can be exceeded with half of the dimensions. Finally, the two-tailed paired T-test comparing Approach



(a) bert-base-multilingual-cased mSTSb fine-tuned version

(b) distilbert-base-multilingual-cased mSTSb fine-tuned version

(c) xlm-roberta-base mSTSb fine-tuned version

(d) xlm-roberta-large mSTSb fine-tuned version

(e) LaBSE mSTSb fine-tuned version

Fig. 4 Approach 4 average Spearman r_s correlation coefficient in multilingual tasks from the mSTSb test as a function of the number of dimensions for the different dimensionality reduction techniques grouped by model

2 vs Approach 4 using the values from Table 4 resulted in $p = 0.255$, revealing no significant difference in performance using these dimensional reduction techniques.

Execution Time Comparison

Although it is out of the main scope of the paper, this subsection compares the different dimension reduction techniques regarding the execution time required to fit each technique. As mentioned in the "Data'' section, it is fundamental to note that KPCA and UMAP were fitted using a subset of $10k$ pairs of sentences due to their computational cost, which is more expensive than their linear counterparts. Therefore, for fair comparisons, IPCA, ICA, and variation threshold techniques are compared on the one hand and KPCA and UMAP on the other.

Finally, we analyse the reduction of computational time by comparing the fitting time of these techniques in Approach 3 and 4 concerning the fine-tuning process performed in Approach 2, following the experimental design described in the "Experimental Setup'' section.

Dimensionality Reduction Techniques Comparison: IPCA, ICA and Variance Threshold

Considering the wide range of dimensions explored for each model, we analyse the fitting time for the minimum, the in-between, and the maximum number of dimensions technique in the mSTSb train split to compare the different techniques in execution time for Approaches 3 and 4. The results of these tests and the dimensions associated with these times are presented in Table 7. It should be noted that the number of dimensions shown in this table for the variance threshold technique differs from the other techniques since the number of reduced dimensions depends on the number of dimensions that exceed the variance threshold explored.

Table 7, shows that the technique that requires the longest fitting time of IPCA, ICA and variance threshold is ICA, while the variance threshold is the one that requires the shortest time.

Regarding computational complexity, IPCA has constant memory complexity $O(bd^2)$ [67], while the ICA algorithm is $O(2md(d+1)n)$ [78], where n is the number of samples of dimensions d, b is the batch size for the IPCA algorithm, and m is the number of iterations of the ICA algorithm. We can observe that when IPCA and ICA are compared for the same number of dimensions and number of samples, the computational burden of the ICA technique is the number of iterations (m) since they critically increase execution time. As stated in [78], this effect is more noticeable as the dimension increases, making the use of ICA computationally unfeasible when the number of dimensions is moderate or high. Our results support these findings, as the ICA's fitting time explodes when the number of dimensions is high (see Table 7). On the other hand, the IPCA execution time remains practically constant because the batches avoid attempting to load all the data into memory at the same time.

Finally, as expected, the feature selection technique is the least time-consuming since it only requires computing the variance for each dimension and filtering it according to a selected variance threshold.

Consequently, there is a trade-off between the number of dimensions to be reduced, the fitting time, and the percentage of performance to be retained. We would recommend that for the case of pre-trained embedding reduction, the ICA technique should be the priority if a low number of dimensions is explored since it is the best at retaining and improving the initial performance at a feasible cost. However, suppose the target number of dimensions to be reduced is medium or high. In that case, we recommend the IPCA technique since it can retain and improve performance with a constant computational cost. Finally, suppose that the execution time is clearly prioritised. In that case, we recommend the variance threshold feature selection technique, even if the number of dimensions is reduced and the performance retained is not the best, as this technique cannot handle the noise in pre-trained features not adjusted for a downstream task.

Dimensionality Reduction Techniques Comparison: KPCA and UMAP

Regarding the KPCA technique, using a kernel function before performing linear PCA increases the execution time. These kernel functions map the data into a nonlinear feature space, so the execution time varies depending on the kernel type. From the results shown in Table 7, the sigmoid kernel requires the most execution time and the cosine kernel the least. Despite these previous results, the selection of the kernel type depends on each case, and different kernels should be explored, as previous studies suggest [62, 65].

Finally, it can be seen from the results shown in Table 7 that the execution time of UMAP is similar to KPCA. Nevertheless, several considerations must be taken into account in the case of the UMAP technique. As discussed in the "Dimensionality reduction Techniques'' section, the number of neighbours parameter is proportional to the amount of global information retained. As shown in Table 2, different values of the number of neighbours are explored for each model. The number of neighbours that provided the best results were 10, 5, 5, 5 and 125 for each model in the same order as the models shown in Table 7. The value of this parameter determines the execution time of the technique; increasing the time, the greater the number of neighbours

to consider for embedding a sentence. For this reason, the degree of similarity between KPCA and UMAP run times varies from one model to another. Similarly, although the table only includes the execution times for the best UMAP parameter values found for each model, the choice of these parameters requires a prior exploration process that can be very long, depending on how intensive the search for the best parameter values is. Hence, this technique has an additional cost to consider.

Fitting Dimensionality Reduction Technique and Fine-tuning Comparison

Another point to discuss is comparing the execution times required by dimension reduction techniques with the fine-tuning process. It should be noted that the fine-tuning times shown in Table 8 do not include the time required for hyperparameter optimization, which, depending on how extensive the search is, can extend the time required to obtain a fine-tuned model. Therefore, the time shown refers to the execution time needed to fine-tune each model once the optimal hyperparameters are found.

In practical application, the time required for fitting a dimensionality reduction technique on pre-trained embeddings instead of fine-tuning the model downstream is what matters. For this purpose, we use the fine-tuning process execution times shown in Table 8 compared to Table 5, which shows the fitting time of the dimension reduction techniques of Approach 3 that improves the result of the baseline Approach 1.

For space reasons, even though Table 5 shows a breakdown of the performance for each technique, the technique used to compare the fine-tuning of each model is the technique that, with the lowest number of dimensions, retains the highest possible performance (highlighted in bold in Table 5). As stated previously, the best technique for all models is ICA. This technique requires a fitting time of an average of $96.68\% \pm 0.68\%$ faster than the fine-tuning process on all models.

Taken together, we can see the advantage in execution time of using reduction techniques in Approach 3 with respect to the fine-tuning process carried out in Approach 2. Remarkably, it is exciting to point out the potential of ICA as an alternative to the fine-tuning process. Although it has previously been proven that fine-tuning achieves better performance than fitting dimension reduction techniques, this technique does not require GPU. This technique reduces the execution time by more than 96%. This fact, together with the ability to improve the performance of pre-trained embeddings and the generalisation capability of this unsupervised technique, reveals the potential of this technique as an alternative to fine-tuning.

UMAP: a Case of Study

In this section, we pay special attention to the recently proposed UMAP technique [66]. The case of UMAP shows that for the STS task, it is not a suitable technique to reduce the dimensionality of the embeddings since it is the one that retains the lowest percentage of the baseline performances in both pre-trained and fine-tuned embeddings (see Tables 5 and 6). Considering this fact, it is interesting to note that the potential of this technique resides in the fact that it quickly saturates, i.e. the maximum retained performance is reached with a small number of dimensions in Approach 3 with an average of $94.65\% \pm 6.07\%$ of reduced initial dimensions retaining $49.00\% \pm 11.14\%$ performance concerning the reference Approach 1, and more notably in Approach 4 with an average of $98.42\% \pm 0.72\%$ of reduced initial dimensions retaining $76.00\% \pm 3.74\%$ performance for the reference Approach 2.

This saturation behaviour can be attributed to the demonstrated functionality of UMAP for generating visualisations by reducing high-dimensional data to 2- or 3-dimensions. As reported in other works [79, 80], the most significant potential of this technique lies precisely in visualisation, where this saturation capability is exploited. UMAP is a nonlinear graph-based method for dimensionality reduction that is not meant to extract features as methods like PCA do [66]. Instead, its primary goal is to represent the input data as a high-dimensional graph and then reconstruct it in a lower-dimensional space while retaining structure. These characteristics explain the lack of good results for the UMAP technique in the mSTSb task in Approach 3, where feature extraction has proven to help manage the noisy pre-trained features. Furthermore, this would explain why we observe UMAP saturation with fewer dimensions in Approach 4, as the embeddings are already fine-tuned, and UMAP can exploit its potential to preserve information.

Previously, it has been proven that its fitting requires an execution time similar to other techniques such as KPCA. It is also important to note that its computational cost limits the total number of data instances to be used for fitting since the first step is expensive. It requires computing a graphical representation of the dataset and then learning an embedding for that graph.

Despite the previous conclusion, UMAP is time-consuming. This technique highly depends on the parameters used, and many parameters can be explored. Additionally, the slow process of embedding new data in a fitted UMAP must also be considered. UMAP does not build a function that directly maps high-dimensional points down to a reduced space. Instead, it first computes a graph representing the whole data and then learns an embedding for it. Thus, embedding a new point

Table 5 Analysis of the lowest reduced number of dimensions from Approach 3 that improves the result of the baseline Approach 1 for a specific performance threshold retained. For instance, 100% threshold represents that the technique achieves at least the 100% of the baseline approach score

Model (Ap. 1 Avg r_s)	Technique	Threshold Performance Retained	Dimensions (% reduction)	Ap. 3 Avg r_s	Fitting Time
bert-base-multilingual-cased (0.4342)	IPCA	100%	209 (73%)	0.4251	27 s
	ICA	**100%**	**89 (88%)**	0.4779	1 m 10 s
	poly	95%	249 (68%)	0.4130	49 s
	rbf	95%	448 (42%)	0.4138	1 m 38 s
	sigmoid	100%	129 (83%)	0.4425	40 s
	cosine	100%	209 (73%)	0.4350	36 s
	UMAP	50%	129 (83%)	0.2176	35 s
	VarThres	85%	161(79%)	0.3727	2 s
distilbert-base-multilingual-cased (0.4531)	IPCA	100%	209 (73%)	0.4553	38 s
	ICA	**100%**	**49 (94%)**	0.4564	43 s
	poly	95%	369 (52%)	0.4310	1 m 6 s
	rbf	95%	608 (21%)	0.4305	2 m 13 s
	sigmoid	100%	129 (83%)	0.4642	38 s
	cosine	100%	209 (73%)	0.4537	33 s
	UMAP	40%	49 (94%)	0.3942	18 s
	VarThres	95%	238 (69%)	0.438	2 s
xlm-roberta-base (0.3274)	IPCA	100%	89 (88%)	0.3711	38 s
	ICA	**100%**	**49 (94%)**	0.4043	56 s
	poly	100%	129 (83%)	0.3439	25 s
	rbf	100%	129 (83%)	0.3439	50 s
	sigmoid	100%	49 (94%)	0.3425	29 s
	cosine	100%	89 (88%)	0.3709	15 s
	UMAP	40%	10 (99%)	0.1320	15 s
	VarThres	100%	52 (93%)	0.3310	2 s
xlm-roberta-large (0.2885)	IPCA	100%	116 (89%)	0.3149	1 m 37 s
	ICA	**100%**	**63 (94%)**	0.3642	1 m 52 s
	poly	100%	223 (78%)	0.2927	45 s
	rbf	100%	276 (73%)	0.2934	1 m 8 s
	sigmoid	100%	63 (94%)	0.2927	32 s
	cosine	100%	116 (89%)	0.3191	25 s
	UMAP	45%	10 (99%)	0.1365	15 s
	VarThres	100%	598 (42%)	0.2917	3 s
LaBSE (0.7096)	IPCA	100%	129 (83%)	0.7251	37 s
	ICA	**100%**	**89 (88%)**	0.7431	1 m 35 s
	poly	100%	169 (78%)	0.7181	34 s
	rbf	100%	408 (47%)	0.7106	1 m 35 s
	sigmoid	100%	89 (88%)	0.7127	34 s
	cosine	100%	129 (83%)	0.7232	21 s
	UMAP	70%	10 (99%)	0.5026	33 s
	VarThres	85%	217 (72%)	0.6148	2 s

requires calculating its nearest neighbours from the fitting data and embedded in the learnt graph. Again, this process execution time increases with the number of neighbours value selected when. Therefore, UMAP can transform new data, albeit more slowly than other techniques that allow this.

Previous Works Comparison

Finally, we discuss our results concerning previous related work in dimensionality reduction of pre-computed embeddings. Before this discussion proceeds, it is essential to

Table 6 Analysis of the lowest reduced number of dimensions from Approach 4 that improves the result of the baseline Approach 2 for a specific performance threshold retained. For instance, 100% threshold represents that the technique achieves at least the 100% of the baseline approach score

Model (Ap. 2 Avg r_s)	Technique	Threshold Performance Retained	Dimensions (% reduction)	Ap. 4 Avg r_s	Fitting Time
bert-base-multilingual-cased-fine-tuned (0.7045)	IPCA	95%	49 (94%)	0.6710	34 s
	ICA	**100%**	**169 (78%)**	0.7047	3 m 37 s
	poly	95%	129 (83%)	0.6716	31 s
	rbf	95%	169 (78%)	0.6738	54 s
	sigmoid	100%	329 (57%)	0.7048	1 m 34 s
	cosine	95%	49 (94%)	0.6707	11 s
	UMAP	70%	10 (99%)	0.5398	32 s
	VarThres	100%	393 (53%)	0.7046	2 s
distilbert-base-multilingual-cased-fine-tuned (0.6863)	IPCA	**95%**	**49 (94%)**	0.6533	35 s
	ICA	**95%**	**49 (94%)**	0.6556	56 s
	poly	95%	129 (83%)	0.6542	30 s
	rbf	95%	129 (83%)	0.6520	49 s
	sigmoid	**95%**	**49 (94%)**	0.6601	29 s
	cosine	95%	89 (88%)	0.6631	16 s
	UMAP	75%	10 (99%)	0.5189	25 s
	VarThres	95%	66 (91%)	0.6620	2 s
xlm-roberta-base-fine-tuned (0.7470)	IPCA	95%	49 (94%)	0.7198	36 s
	ICA	95%	49 (94%)	0.7208	59 s
	poly	95%	89 (88%)	0.7112	25 s
	rbf	95%	129 (83%)	0.7134	49 s
	sigmoid	**100%**	**289 (62%)**	0.7472	1 m 45 s
	cosine	95%	49 (94%)	0.7195	11 s
	UMAP	75%	10 (99%)	0.5724	31 s
	VarThres	100%	411 (46%)	0.7491	3 s
xlm-roberta-large-fine-tuned (0.8150)	IPCA	95%	63 (94%)	0.7910	51 s
	ICA	95%	63 (94%)	0.7950	1 m 16 s
	poly	95%	63 (94%)	0.7774	23 s
	rbf	95%	63 (94%)	0.7760	42 s
	sigmoid	**100%**	**223 (78%)**	0.8151	50 s
	cosine	95%	63 (94%)	0.7916	13 s
	UMAP	80%	10 (99%)	0.6584	38 s
	VarThres	95%	95 (91%)	0.7936	3 s
LaBSE-fine-tuned (0.8242)	IPCA	95%	89 (88%)	0.8014	34 s
	ICA	95%	89 (88%)	0.7986	2 m 3 s
	poly	95%	89 (88%)	0.7898	25 s
	rbf	95%	129 (83%)	0.7932	47 s
	sigmoid	**100%**	**728 (5%)**	0.8243	7 m 11 s
	cosine	95%	89 (88%)	0.8001	21 s
	UMAP	80%	23 (97%)	0.6640	35 s
	VarThres	95%	227 (70%)	0.7964	2 s

remark that our work differs from previous work in the literature as we explore a more comprehensive range of unsupervised dimension reduction techniques, evaluating them on pre-trained and fine-tuned pre-computed embeddings from state-of-the-art multilingual contextual-based transformer models in the Semantic Textual Similarity (STS) task.

Our experimental results are in agreement with Raunak et al. [21, 22], Singh et al. [27] and Thirumoorthy and Muneeswaran [37], which corroborates the hypothesis that reduction in pre-trained embeddings can maintain or improve the performance of the original embeddings. Similarly, our results are consistent with those obtained by

Table 7 Time required to fit the different dimension reduction techniques in the mSTSb task. Owing to the fact that a wide range of dimensions is explored, the fitting time for the minimum, in-between and maximum number of dimensions are reported for each model and technique. The measurements are performed on an Intel(R) Xeon(R) Bronze 3206R CPU at 1.90GHz

Model	Technique	Time Min Dimension	Time In-between Dimension	Time Max Dimension
bert-base-multilingual-cased	IPCA	27 s (10)	29 s (448)	30 s (768)
	ICA	53 s (10)	8 m 43 s (448)	52 m 48 s (768)
	poly	18 s (10)	1 m 26 s (448)	2 m 17 s (768)
	rbf	36 s (10)	1 m 38 s (448)	3 m 29 s (768)
	sigmoid	24 s (10)	2 m 15 s (448)	5 m 53 s (768)
	cosine	8 s (10)	55 s (408)	1 m 24 s (768)
	UMAP	20 s (10)	1 m 30 s (448)	3 m 29 s (768)
	VarThres	2 s (65)	2 s (516)	2 s (767)
distilbert-base-multilingual-cased	IPCA	36 s (10)	40 s (448)	44 s (768)
	ICA	29 s (10)	8 m 30 s (448)	29 m 31 s (768)
	poly	16 s (10)	1 m 24 s (448)	2 m 35 s (768)
	rbf	35 s (10)	1 m 44 s (448)	2 m 57 s (768)
	sigmoid	23 s (10)	3 m 58 s (448)	15 m 40 s (768)
	cosine	8 s (10)	51 s (448)	1 m 51 s (768)
	UMAP	13 s (10)	1 m 38 s (448)	4 m 19 s (768)
	VarThres	2 s (66)	2 s (507)	2 s (767)
xlm-roberta-base	IPCA	35 s (10)	35 s (448)	40 s (768)
	ICA	56 s (10)	9 m 54 s (448)	21 m 9 s (768)
	poly	16 s (10)	1 m 2 s (448)	2 m 12 s (768)
	rbf	33 s (10)	1 m 38 s (448)	2 m 2 s (768)
	sigmoid	23 s (10)	1 m 43 s (448)	3 m 26 s (768)
	cosine	7 s (10)	56 s (448)	1 m 36 s (768)
	UMAP	15 s (10)	2 m (448s)	4 m 32 s (768)
	VarThres	2 s (1)	2 s (448)	2 s (768)
xlm-roberta-large	IPCA	54 s (10)	54 s (490)	1 m 46 s (1024)
	ICA	57 s (10)	23 m 13 s (490)	32 m 15 s (1024)
	poly	16 s (10)	1 m 24 s (490)	3 m 35 s (1024)
	rbf	35 s (10)	1 m 30 s (490)	4 m 49 s (1024)
	sigmoid	23 s (10)	2 m 58 s (490)	8 m 28 s (1024)
	cosine	9 s (10)	1 m 38 s (490)	2 m 41 s (1024)
	UMAP	15 s (10)	1 m 57 s (490)	7 m 24 s (1024)
	VarThres	3 s (10)	3 s (598)	3 s (1024)
LaBSE	IPCA	35 s (10)	36 s (448)	39 s (768)
	ICA	36 s (10)	12 m 55 s (448)	52 m 39 s (768)
	poly	17 s (10)	1 m 24 s (448)	2 m 35 s (768)
	rbf	35 s (10)	1 m 41 s (448)	2 m 50 s (768)
	sigmoid	24 s (10)	4 m 25 s (448)	19 m 37 s (768)
	cosine	8 s (10)	55 s (448)	1 m 44 s (768)
	UMAP	33 s (10)	2 m 11 s (448)	4 m 18 s (768)
	VarThres	2 s (94)	2 s (615)	2 s (767)

Truşcă et al. [23] and Shimomoto et al. [47]. The authors employ PCA as the only unsupervised technique. We have confirmed that this feature extraction technique has great potential to reduce the dimensions of pre-trained embeddings. Our results further reveal that incremental PCA (IPCA) is also suitable for embedding reduction. Besides, we found that for embeddings already fine-tuned to the downstream task, using the KPCA, the nonlinear version of PCA, is much more helpful for preserving performance and reducing dimensions.

Regarding execution time, our results are similar to Saeed et al. [17]. These authors combined the unsupervised PCA technique and the supervised LDA technique with classical NLP techniques based on N-grams and classical ML models (Decision Trees, Logistic Regression, or Naive Bayes) for sentiment classification of monolingual Arabic texts.

Table 8 Time required to fine-tune the different models in the mSTSb task. The fine-tuning process is carried out in Quadro-RTX 8000 GPU. The best hyperparameters configuration considered for each model is also reported

Model	Time	Hyperparameters
bert-base-multilingual-cased-fine-tuned	39 m 18 s	bach_size: 32
		lr: 2e-5
		epochs: 2
		scheduler: warmuplinear
		warmup_ratio: 0.2
		wieght_decay: 0.2
distilbert-base-multilingual-cased-fine-tuned	15 m 31 s	bach_size: 64
		lr: 2e-5
		epochs: 2
		scheduler: warmuplinear
		warmup_ratio: 0.3
		wieght_decay: 0.7
xlm-roberta-base-fine-tuned	27 m 49 s	bach_size: 64
		lr: 5e-5
		epochs: 2
		scheduler: warmuplinear_hard_restarts
		warmup_ratio: 0.1
		wieght_decay: 0.5
xlm-roberta-large-fine-tuned	1 h 2 m 32 s	bach_size: 64
		lr: 1e-5
		epochs: 2
		scheduler: warmupcosine
		warmup_ratio: 0.2
		wieght_decay: 0
LaBSE-fine-tuned	59 m 39 s	bach_size: 32
		lr: 3e-6
		epochs: 2
		scheduler: warmupcosine
		warmup_ratio: 0.1
		wieght_decay: 0.5

The authors reported an improvement in results over previous work by reducing dimensions by up to 93% with a 97% shorter run time.

Even though the NLP task addressed in the present study is STS from a multilingual perspective (including Arabic) and not sentiment classification, our findings align with these previous results. The ICA technique reduces an average of $91.58\% \pm 2.59\%$ of the initial dimensions, retaining 100% of the baseline Approach 1 performance, requiring a fitting time of $96.68\% \pm 0.68\%$ faster than the fine-tuning process.

Given these points, we can safely conclude that our results are in line with previous work extending their findings to state-of-the-art contextual-based models from a multilingual approach. This paper provides new insights into dimensionality reduction techniques for a space- and time-efficient data representation.

Conclusion

In this investigation, the goal was to assess the impact of a variety of dimensionality reduction techniques on the performance of pre-computed multilingual siamese fashion transformers embeddings on semantic textual similarity tasks from mSTSb, to expand our knowledge of semantic-aware transformer-based models. To this end, two different baseline approaches are reduced (i.e. Approach 1 and Approach 2), one using the pre-trained version of the models and the second further fine-tuning them on the downstream STS task. Particular attention is paid to analysing which techniques best and with the fewest dimensions improve the performance of the baseline approaches.

From the research, it is possible to conclude that dimensionality reduction techniques can help reduce the number of dimensions of the embeddings while improving the results

if using pre-trained embeddings from Approach 1 and preserving the performance when using fine-tuned embeddings from Approach 2. Nevertheless, the dimensionality reduction is more considerable in the pre-trained version with an average of 91.58% ± 2.59% compared to the average of 54.65% ± 32.20% of the fine-tuned version. Special attention is given to ICA in the pre-trained scenario, which adequately managed the noisy variables present in not adjusted embeddings. This technique also proved to be a reasonable alternative to fit the models in the downstream task in an unsupervised way, leading to a generalised adjusted version of the models with downstream multitasking capabilities. Nevertheless, it has also been proved that this unsupervised fitting is not comparable to supervised fine-tuning. On the other hand, the fine-tuned scenario revealed the relevance of feature selection techniques and the significance of nonlinear KPCA techniques for dimensionality reduction.

Additionally, although our execution time experiments reveal that ICA is the most time-consuming technique among the dimension reduction techniques, reducing pre-trained embeddings (Approach 3) by gaining generalisation power with this unsupervised technique is still faster than performing downstream fine-tuning (Approach 2). Moreover, ICA does not require GPUs for fitting, reinforcing its potential as an alternative to fine-tuning. Overall, a good trade-off between performance against available computational resources, execution time, and the number of dimensions to be reduced must be considered when choosing the approach to follow.

The results of our experiments are consistent with previous results from the literature, corroborating the hypothesis that reduction in embeddings can maintain or improve the performance of the original embeddings by extending their evaluation to state-of-the-art contextual-based models from a multilingual approach. In this way, we can establish that dimensionality reduction techniques could also be leveraged for contextualised embeddings.

To our knowledge, this is the first study to investigate the effect of dimensionality reduction techniques and transformers models in a multilingual semantic-awareness scenario. This study analyses alternatives to the storage limitation we are about to face if the current trends of using large datasets and the growth rate of storage utilisation persist. Based on the promising findings presented in this article, continued research into the impact of dimensionality reduction techniques in other highly demanded NLP tasks appears to be totally justified. Furthermore, in future research, we intend to focus on testing the reduced models presented in this work in real-world applications. Besides, we hope to carry out further experimental investigation, including other dimensionality reduction approaches, such as creating a distilled version of pre-trained models or exploring the novel feature extraction and feature selection methods proposed in [27, 37].

As stated previously, the findings presented in this study of multilingual semantic similarity are of direct practical applicability. Combining dimensionality reduction techniques with transformer models could also help reduce the embedding size and make ensemble approaches possible. Finally, further studies about the multitasking generalisation capabilities of ICA for pre-trained models are still required.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This research has been supported by the Spanish Ministry of Science and Education under FightDIS (PID2020-117263GB-100), by MCIN/AEI/10.13039/501100011033/ and European Union NextGenerationEU/PRTR for XAI-Disinfodemics (PLEC2021-007681) grant, by Comunidad Autónoma de Madrid under S2018/ TCS-4566 (CYNAMON) grant, by BBVA Foundation grants for scientific research teams SARS-CoV-2 and COVID-19 under the grant: "*CIVIC: Intelligent characterisation of the veracity of the information related to COVID-19*", and by IBERIFIER (Iberian Digital Media Research and Fact-Checking Hub), funded by the European Commission under the call CEF-TC-2020-2, grant number 2020-EU-IA-0252. Finally, David Camacho has been supported by the Comunidad Autónoma de Madrid under Convenio Plurianual with the Universidad Politécnica de Madrid in the actuation line of "*Programa de Excelencia para el Profesorado Universitario*" and by the research project DisTrack: Tracking disinformation in Online Social Networks through Deep Natural Language Processing, granted by Barcelona Mobile World Capital Foundation.

Data Availability The Multilingual Semantic Textual Similarity Benchmark (mSTSb) used for the purpose of this article can be found in https://github.com/Huertas97/Multilingual-STSB.

Declarations

Research Involving Human Participants and/or Animals This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

Conflict of Interest The authors declare no competing interests.

References

1. Otter DW, Medina JR, Kalita JK. A survey of the usages of deep learning for natural language processing. IEEE Trans Neural Netw Learn Syst. 2021;32(2):604–24. https://doi.org/10.1109/TNNLS.2020.2979670.
2. Tay Y, Dehghani M, Bahri D, Metzler D. Efficient transformers: a survey. ACM Computing Surveys. 2022. https://doi.org/10.1145/3530811.
3. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All You Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 6000–10. https://doi.org/10.5555/3295222.3295349.
4. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 4171–86. https://doi.org/10.18653/v1/N19-1423.
5. Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics; 2019. p. 3982–92. https://doi.org/10.18653/v1/D19-1410.
6. Huertas-Tato J, Martin A, Camacho D. BERTuit: Understanding Spanish language in Twitter through a native transformer. 2022. https://doi.org/10.48550/ARXIV.2204.03465.
7. Chowdhary KR. Natural language processing. New Delhi: Springer India; 2020. p. 603–49. https://doi.org/10.1007/978-81-322-3972-7_19.
8. Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L. SemEval-2017 Task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics; 2017. p. 1–14. https://doi.org/10.18653/v1/S17-2001.
9. Humeau S, Shuster K, Lachaux MA, Weston J. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In: International Conference on Learning Representations (ICLR). Online, 2020. https://doi.org/10.48550/ARXIV.1905.01969.
10. Zhelezniak V, Savkov A, Shen A, Hammerla N. Correlation coefficients and semantic textual similarity. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 951–62. https://doi.org/10.18653/v1/N19-1100.
11. Sidorov G, Gelbukh A, Gómez-Adorno H, Pinto D. Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas. 2014;18(3):491–504. https://doi.org/10.13053/cys-18-3-2043.
12. Cambria E, Wang H, White B. Guest editorial: Big social data analysis. Knowl Based Syst. 2014;69:1–2. https://doi.org/10.1016/j.knosys.2014.07.002.
13. Araque O, Corcuera-Platas I, Sánchez-Rada JF, Iglesias CA. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Exp Syst App. 2017;77:236–46. https://doi.org/10.1016/j.eswa.2017.02.002.
14. Zhou Y, Yang Y, Liu H, Liu X, Savage N. Deep learning based fusion approach for hate speech detection. IEEE Access. 2020;8:128923–9. https://doi.org/10.1109/ACCESS.2020.3009244.
15. Khan A, Sohail A, Zahoora U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks. Artif Intell Rev. 2020;53(8):5455–516. https://doi.org/10.1007/s10462-020-09825-6.
16. Chau EC, Smith NA. Specializing multilingual language models: an empirical study. In: Proceedings of the 1st Workshop on Multilingual Representation Learning. Punta Cana, Dominican Republic: Association for Computational Linguistics; 2021. p. 51–61. https://doi.org/10.18653/v1/2021.mrl-1.5.
17. Saeed RMK, Rady S, Gharib TF. Optimizing sentiment classification for Arabic opinion texts. Cognit Comput. 2021;13(1):164–78. https://doi.org/10.1007/s12559-020-09771-z.
18. Herbelot A, Zhu X, Palmer A, Schneider N, May J, Shutova E, editors. Proceedings of the Fourteenth Workshop on Semantic Evaluation. Barcelona (online): International Committee for Computational Linguistics; 2020.
19. Ferro N. What happened in CLEF... for a while? In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Cham: Springer International Publishing; 2019. p. 3–45. https://doi.org/10.1007/978-3-030-28577-7_1
20. Introducing the World's Largest Open Multilingual Language Model: BLOOM. 2022. Available from: https://bigscience.huggingface.co/blog/bloom.
21. Raunak V, Gupta V, Metze F. Effective dimensionality reduction for word embeddings. In: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). Florence, Italy: Association for Computational Linguistics; 2019. p. 235–43. https://doi.org/10.18653/v1/W19-4328.
22. Raunak V, Kumar V, Gupta V, Metze F. On dimensional linguistic properties of the word embedding space. In: Proceedings of the 5th Workshop on Representation Learning for NLP. Online: Association for Computational Linguistics; 2020. p. 156–65. https://doi.org/10.18653/v1/2020.repl4nlp-1.19.
23. Truşcă MM, Aldea A, Grădinaru SE, Albu C. Post-processing and dimensionality reduction for extreme learning machine in text classification. Econ Comput Econ Cybern Stud Res. 2021;55(4):37–50. https://doi.org/10.24818/18423264/55.4.21.03.
24. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. J Am Soc Info Sci. 1990;41(6):391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.
25. Sun W, Du Q. Hyperspectral band selection: a review. IEEE Geosci Remote Sens Mag. 2019;7(2):118–39. https://doi.org/10.1109/MGRS.2019.2911100.
26. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A review of unsupervised feature selection methods. Artif Intell Rev. 2020;53(2):907–48. https://doi.org/10.1007/s10462-019-09682-y.
27. Singh KN, Devi SD, Devi HM, Mahanta AK. A novel approach for dimension reduction using word embedding: an enhanced text classification approach. Int J Info Manage Data Insights. 2022;2(1):100061. https://doi.org/10.1016/j.jjimei.2022.100061.
28. Maxwell AE, Warner TA, Fang F. Implementation of machine-learning classification in remote sensing: an applied review. Int J Remote Sens. 2018;39(9):2784–817. https://doi.org/10.1080/01431161.2018.1433343.
29. Patel AA. Hands-on unsupervised learning using Python: How to build applied machine learning solutions from unlabeled data. Sebastopol, California: O'Reilly; 2019.
30. Hira ZM, Gillies DF. A review of feature selection and feature extraction methods applied on microarray data. Adv Bioinformatics. 2015;2015:198363–13. https://doi.org/10.1155/2015/198363.

31. Xu D, Yen IEH, Zhao J, Xiao Z. Rethinking network pruning – under the pre-train and fine-tune paradigm. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics; 2021. p. 2376–82. https://doi.org/10.18653/v1/2021.naaclmain.188.
32. Bahdanau D, Bosc T, Jastrzebski S, Grefenstette E, Vincent P, Bengio Y. Learning to compute word embeddings on the fly. 2017. https://doi.org/10.48550/ARXIV.1706.00286.
33. Johnson J, Douze M, Jégou H. Billion-scale similarity search with GPUs. IEEE Trans Big Data. 2021;7(3):535–47. https://doi.org/10.1109/TBDATA.2019.2921572.
34. Mitra B, Craswell N. An introduction to neural information retrieval. Foundations and Trends® in Information Retrieval. 2018;13(1):1–126. https://doi.org/10.1561/1500000061.
35. Camastra F, Vinciarelli A. Feature extraction methods and manifold learning methods. In: Machine Learning for Audio, Image and Video Analysis. London: Springer London; 2008. p. 305–41. https://doi.org/10.1007/978-1-84800-007-0_11.
36. Egger R. In: Egger R, editor. Text representations and word embeddings. Cham: Springer International Publishing; 2022. p. 335–61. https://doi.org/10.1007/978-3-030-88389-8_16.
37. Thirumoorthy K, Muneeswaran K. Feature selection for text classification using machine learning approaches. Natl Acad Sci Lett. 2022;45(1):51–6. https://doi.org/10.1007/s40009-021-01043-0.
38. Strubell E, Ganesh A, McCallum A. Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics; 2019. p. 3645–50. https://doi.org/10.18653/v1/P19-1355.
39. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013;35(8):1798–828. https://doi.org/10.1109/TPAMI.2013.50.
40. Choi SW, Kim BHS. Applying PCA to deep learning forecasting models for predicting PM2.5. Sustainability. 2021;13(7). https://doi.org/10.3390/su13073726.
41. Menaga D, Revathi S. Probabilistic Principal Component Analysis (PPCA) based dimensionality reduction and deep learning for cancer classification. In: Dash SS, Das S, Panigrahi BK, editors. Intell Comput Appl. Singapore: Springer Singapore; 2021. p. 353–68. https://doi.org/10.1007/978-981-15-5566-4_31.
42. Kushwaha N, Pant M. Textual data dimensionality reduction - a deep learning approach. Multimedia Tools Appl. 2020;79(15–16):11039–50. https://doi.org/10.1007/s11042-018-6900-x.
43. Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics; 2014. p. 1532–43. https://doi.org/10.3115/v1/D14-1162.
44. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Trans Assoc Comput Linguistics. 2017;5:135–46. https://doi.org/10.1162/tacl_a_00051.
45. Pearson K. On lines and planes of closest fit to systems of points in space. London Edinburgh Dublin Philos Mag J Sci. 1901;2(11):559–72. https://doi.org/10.1080/14786440109462720.
46. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Philos Trans Royal Soc Math Phys Eng Sci. 2016;374(2065). https://doi.org/10.1098/rsta.2015.0202.
47. Shimomoto EK, Portet F, Fukui K. Text classification based on the word subspace representation. Pattern Anal Appl: PAA. 2021;24(3):1075–93. https://doi.org/10.1007/s10044-021-00960-6.
48. Song H, Zou D, Hu L, Yuan J. Embedding compression with right triangle similarity transformations. In: Artificial Neural Networks and Machine Learning - ICANN 2020. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2020. p. 773–85. https://doi.org/10.1007/978-3-030-61616-8_62.
49. Choudhary R, Doboli S, Minai AA. A comparative study of methods for visualizable semantic embedding of small text corpora. In: 2021 International Joint Conference on Neural Networks (IJCNN); 2021. p. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534250.
50. Hinton G, Roweis S. Stochastic neighbor embedding. In: Proceedings of the 15th International Conference on Neural Information Processing Systems. NIPS'02. Cambridge, MA, USA: MIT Press; 2002. p. 857–64.
51. Huertas-García Á, Huertas-Tato J, Martín A, Camacho D. Countering misinformation through semantic-aware multilingual models. In: Intelligent Data Engineering and Automated Learning – IDEAL 2021. Cham: Springer International Publishing; 2021. p. 312–23. https://doi.org/10.1007/978-3-030-91608-4_31.
52. Nogueira R, Jiang Z, Pradeep R, Lin J. Document ranking with a pretrained sequence-to-sequence model. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics; 2020. p. 708–18. https://doi.org/10.18653/v1/2020.findings-emnlp.63.
53. Robertson S, Zaragoza H, Taylor M. Simple BM25 extension to multiple weighted fields. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management. CIKM '04. New York, NY, USA: Association for Computing Machinery; 2004. p. 42–9. https://doi.org/10.1145/1031171.1031181.
54. Wardle C, Derakhshan H. Information disorder: Toward an interdisciplinary framework for research and policy making. Council of Europe; 2017. Available from: https://rm.coe.int/information-disorder-toward-an-interdisciplinary-frameworkfor-researc/168076277c.
55. Carmi E, Yates SJ, Lockley E, Pawluczuk A. Data citizenship: Rethinking data literacy in the age of disinformation, misinformation, and malinformation. Internet Policy Rev. 2020;9(2). https://doi.org/10.14763/2020.2.1481.
56. Gaglani J, Gandhi Y, Gogate S, Halbe A. Unsupervised WhatsApp fake news detection using semantic search. In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS); 2020. p. 285–9. https://doi.org/10.1109/ICICCS48265.2020.9120902.
57. Huertas-García Á, Huertas-Tato J, Martín A, Camacho D. CIVIC-UPM at CheckThat!2021: Integration of transformers in misinformation detection and topic classification. In: Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum. vol. 2936 of CEUR Workshop Proceedings. Bucharest, Romania: CEUR-WS.org; 2021. p. 520–30.
58. Martín A, Huertas-Tato J, Huertas-García Á, Villar-Rodríguez G, Camacho D. FacTeR-Check: Semi-automated fact-checking through semantic similarity and natural language inference. Knowl Based Syst. 2022;251:109265. https://doi.org/10.1016/j.knosys.2022.109265.
59. Grootendorst M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv: arXiv:2203.05794 [Preprint]. 2022.
60. Grootendorst M. KeyBERT: Minimal keyword extraction with BERT. Zenodo; 2020. https://doi.org/10.5281/zenodo.4461265.
61. Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics; 2020. p. 4512–25. https://doi.org/10.18653/v1/2020.emnlp-main.365.
62. Muller KR, Mika S, Ratsch G, Tsuda K, Scholkopf B. An introduction to kernel-based learning algorithms. IEEE Trans Neural Netw. 2001;12(2):181–201. https://doi.org/10.1109/72.914517.
63. Ross DA, Lim J, Lin RS, Yang MH. Incremental learning for robust visual tracking. Int J Comput Vis. 2007;77(1–3):125–41. https://doi.org/10.1007/s11263-007-0075-7.
64. Hyvärinen A. Independent component analysis: Recent advances. Philos Trans Royal Soc A Math Phys Eng Sci. 2013;371(1984):20110534. https://doi.org/10.1098/rsta.2011.0534.

65. Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a Kernel Eigenvalue problem. Neural Comput. 1998;10(5):1299–319. https://doi.org/10.1162/089976698300017467.
66. McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform Manifold Approximation and Projection. J Open Source Softw. 2018;3(29):861. https://doi.org/10.21105/joss.00861.
67. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011;12:2825–30. https://doi.org/10.48550/ARXIV.1201.0490.
68. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics; 2020. p. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6.
69. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv: arXiv:1910.01108 [Preprint]. 2019.
70. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised cross-lingual representation learning at scale. 2019. https://doi.org/10.48550/ARXIV.1911.02116.
71. Liu Z, Lin W, Shi Y, Zhao J. A robustly optimized BERT pre-training approach with post-training. In: Chinese Computational Linguistics: 20th China National Conference, CCL 2021, Hohhot, China, August 13-15, 2021, Proceedings. Berlin, Heidelberg: Springer-Verlag; 2021. p. 471–84. https://doi.org/10.1007/978-3-030-84186-7_31.
72. Feng F, Yang Y, Cer D, Arivazhagan N, Wang W. Language-agnostic BERT sentence embedding. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. vol.1. Dublin, Ireland: Association for Computational Linguistics; 2022. p. 878–91. https://doi.org/10.18653/v1/2022.acl-long.62.
73. Reimers N, Beyer P, Gurevych I. Task-oriented intrinsic evaluation of semantic textual similarity. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: The COLING 2016 Organizing Committee; 2016. p. 87–96.
74. Wang A, Singh A, Michael J, Hill F, Levy O, Bowman S. GLUE: a multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Brussels, Belgium: Association for Computational Linguistics; 2018. p. 353–5. https://doi.org/10.18653/v1/W18-5446.
75. Bishop CM. Pattern recognition and machine learning (information science and statistics). Berlin, Heidelberg: Springer-Verlag; 2006.
76. Liu C. Enhanced independent component analysis and its application to content based face image retrieval. IEEE Trans Syst Man Cybern B - Cybern. 2004;34(2):1117–27. https://doi.org/10.1109/TSMCB.2003.821449.
77. Ekenel HK, Sankur B. Multiresolution face recognition. Image Vis Comput. 2005;23(5):469–77. https://doi.org/10.1016/j.imavis.2004.09.002.
78. Laparra V, Camps-Valls G, Malo J. Iterative Gaussianization: From ICA to random rotations. IEEE Trans Neural Netw. 2011;22(4):537–49. https://doi.org/10.1109/TNN.2011.2106511.
79. Cao J, Spielmann M, Qiu X, Huang X, Ibrahim DM, Hill AJ, et al. The single-cell transcriptional landscape of mammalian organogenesis. Nature. 2019;566(7745):496. https://doi.org/10.1038/s41586-019-0969-x.
80. Carter S, Armstrong Z, Schubert L, Johnson I, Olah C. Activation atlas. Distill. 2019. https://doi.org/10.23915/distill.00015.

4.2 Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflage

This instance is the accepted paper via open access that can be accessed in [224].

Huertas-García, Álvaro, et al. «Countering Malicious Content Moderation Evasion in Online Social Networks: Simulation and Detection of Word Camouflage». Applied Soft Computing, vol. 145, 2023, p. 110552. DOI.org (Crossref), https://doi.org/10.1016/j.asoc.2023.110552

Applied Soft Computing 145 (2023) 110552

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Conference

Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflage

Álvaro Huertas-García *, Alejandro Martín, Javier Huertas-Tato, David Camacho

Department of Computer Systems Engineering, Universidad Politécnica de Madrid, Madrid, Spain

ARTICLE INFO

Article history:
Received 7 January 2023
Received in revised form 8 June 2023
Accepted 9 June 2023
Available online 30 June 2023

Keywords:
Information disorders
Leetspeak
Word camouflage
Multilingualism
Content evasion

ABSTRACT

Content moderation is the process of screening and monitoring user-generated content online. It plays a crucial role in stopping content resulting from unacceptable behaviors such as hate speech, harassment, violence against specific groups, terrorism, racism, xenophobia, homophobia, or misogyny, to mention some few, in Online Social Platforms. These platforms make use of a plethora of tools to detect and manage malicious information; however, malicious actors also improve their skills, developing strategies to surpass these barriers and continuing to spread misleading information. Twisting and camouflaging keywords are among the most widely used techniques to evade platform content moderation systems. In response to this recent ongoing issue This paper presents an innovative approach to address this linguistic trend in social networks through the simulation of different content evasion techniques and a multilingual transformer model for content evasion detection. In this way a multilingual public tool Named "*pyleetspeak*" is shared with the scientific community, enabling the generation and simulation of content evasion through automatic word camouflage in a customizable way. Additionally a multilingual named-entity recognition (NER) transformer-based model is provided Designed for the recognition and detection of such evasion technique. The developed tool is multilingual Supporting over 20 languages (ar, az, da, de, el, en, es, fi, fr, hu, id, it, kk, nb, ne, nl, pt, ro, ru, sl, sv, tg, tr) and the NER model has been tested in English, Spanish, French, Italian, and German. This multilingual NER model is evaluated in different textual scenarios Detecting different types and mixtures of camouflage techniques Achieving an overall weighted F1 score of 0.8795. This article contributes significantly to countering malicious information by developing multilingual tools to simulate and detect new methods of evasion of content on social networks Making the fight against information disorders more effective

1. Introduction

Regardless of the communication actions between users and their multimedia content, almost every social network today deals with malicious information (that is, misinformation, disinformation, misleading information or any other kind of information pollution) and the ostensible polarization of media discourse [1].

Content moderation has become one of the main ways social media platforms manage this situation. Moderation of content is critical to maintaining a safe and welcoming online environment. It involves reviewing and removing content that violates the rules and policies of a website or platform [2]. However, as these policies improve and new techniques are employed to monitor their compliance, so does how users can evade content moderation efforts [3]. This effect entails severe consequences for both the platform and its users. If content moderation evasion is not addressed adequately, it can lead to the spread of harmful or illegal content, which can damage the reputation of the platform and put its users at risk [4,5]. Therefore, it is essential to provide these platforms with effective strategies to combat content moderation evasion.

In 2016, Facebook started an initiative against false claims pointing out those disproved by fact-checkers [6]. Later in 2020, with the emergence of the coronavirus, Twitter applied similar actions to manage the overabundance of dis/misinformation related to the COVID-19 pandemic, highlighting tweets that were considered to deliberately disseminate incorrect information to undermine public health [7–9]. Additionally, since the origin of the pandemic, Twitter has facilitated developers' and researchers' access to their conversational content, providing a specific COVID-19 streaming endpoint [10] and an academic version of their API without timeline limitations [11], allowing a better study

* Corresponding author.
E-mail addresses: alvaro.huertas.garcia@upm.es (Á. Huertas-García), alejandro.martin@upm.es (A. Martín), javier.huertas.tato@upm.es (J. Huertas-Tato), david.camacho@upm.es (D. Camacho).

https://doi.org/10.1016/j.asoc.2023.110552
1568-4946/

of the spread of hoaxes [12]. Due to the same situation, the YouTube video platform recently established moderation policies for removing videos containing COVID-19 misinformation and limiting recommendations for anti-vaccination videos [13].

Undoubtedly, the COVID-19 pandemic has raised the importance of fighting information disorders and moderating the content published on social networks. Moreover, content filtering has also received critical attention in other fields such as terrorism, hate speech, misogyny, and sexism [3,5]. For example, in 2017, Facebook, Google, Twitter, and Microsoft created the Global Internet Forum to Counter Terrorism (GIFCT) group, which collaborates with the European Commission to combat illegal online hate speech [14]. Instagram, Pinterest, and Tumblr are committed to limiting the results of searches with hashtags related to eating disorders and sexual abuse, among others [3,5].

Nevertheless, malicious actors present on these platforms are aware of these content-moderation rules. Recently, to evade this content filtering, it has been demonstrated that these actors twist and camouflage critical parts of speech using different techniques such as leetspeak [5,15] (which involves generating visually similar character strings by replacing alphabet characters with other symbols), word inversion or inserting punctuation characters into words, procedures belonging to the named word camouflaging [16]. These evasion methods threaten the capabilities of the platform's systems and allow malicious actors to continue spreading falsehood, misleading, and harmful content, as shown below in Section 2.2.

The primary purpose of this study is to provide new instruments in response to these rapid changes in content moderation evasion. The following are the new contributions of our research concerning previous studies.

- A novel methodology is presented and shared with the rest of the scientific community a novel methodology to generate/simulate the content evasion phenomenon from a multilingual level in a customizable way. For the sake of reproducibility, the methodology approach is presented as a public Python package named "pyleetspeak".[1] It should be noted that the tool is customizable and is not language dependent as it supports over 20 languages.[2]
- A curated synthetic multilingual dataset[3] of camouflaged words with applied quality filters is presented. The dataset is shared in different formats to facilitate its applicability to other researchers. The languages considered are English, Spanish, French, Italian, and German.
- A multilingual Transformer-based model[4] is derived to detect and discern different word camouflage techniques to prevent content evasion.
- The potential of multiclass camouflage Named Entity Recognition at the multilingual level is evaluated by comparing the developed multilingual model with monolingual baseline models for the different languages considered.
- Lastly, previous research [17] is continued by reaffirming the usefulness of multilingual pre-training in semantic similarity (mSTSb) to increase the generalizability of multilingual models.

In order to guide the research and achieve the aforementioned contributions, a set of research questions has been formulated to serve as the foundation for the investigation. By addressing these questions, the aim is to develop effective tools and methodologies for content evasion detection and prevention.

- Main RQ - How can a customizable, multilingual methodology be developed to effectively simulate, detect, and prevent content evasion through word camouflage techniques?
- RQ1: How can a novel, customizable methodology for generating and simulating multilingual content evasion via word camouflage be designed, and how can a curated synthetic multilingual dataset be created to support research in content evasion detection?
- RQ2: How can a multilingual Transformer-based model be developed to detect and discern different word camouflage techniques in order to prevent content evasion, and how does it compare with monolingual baseline models across various languages?
- RQ3: Does multilingual pre-training in semantic similarity (mSTSb) continue to contribute to increased generalizability in multilingual models as shown in previous research [17]?

This paper is organized as follows. Section 2 begins by examining previous work on content moderation and content moderation evasion techniques. Section 3 introduces the new methodology to simulate content moderation evasion and generate annotated data, addressing RQ1. Section 4 explains the data used to generate word camouflage and train multilingual word camouflage NER models, while Section 5 discusses the experimental results related to RQ2 and RQ3. Finally, Section 6 covers the main research question and the conclusions drawn from the study.

2. Literature review

2.1. Content moderation

Content moderation is the process of screening and monitoring user-generated content online to suppress communications that are deemed undesirable. As Gerrard & Thornham [18] have highlighted at present, there are two dominant forms of social media content moderation: automated and human. However, the growing amount of content uploaded to social media platforms makes it impossible to rely exclusively on the human content moderation approach [19].

Traditional practices to limit the disruption that can be caused by antisocial behavior consist of blocking messages based on basic text properties (e.g., length), interaction parameters (e.g., posting frequency, reply frequency), or according to the standards of designated moderators [20]. These practices had some drawbacks, such as their applicability to small, medium-sized conversations. These initial practices led to more sophisticated and scalable forms of automated moderation [21,22].

Although these systems from online content platforms remain opaque and poorly understood, two different methods are known. In some cases, automated systems employ fingerprinting or hash matching techniques to compare new content against known data and databases of unwanted or flagged content [18,23]. In other cases, automated systems will be machine learning systems trained on large datasets to spot new or previously unseen allegedly illegal material and remove it, block it, or filter it, which can also be used to create more training data or assist in human moderation [21,22,24]. Consequently, these automated content filtering methods benefit from data sharing. For example, the four members of GIFCT share best practices and databases to develop their automated systems. An example of the importance of this data sharing is the Christchurch attack in New Zealand in 2019, where a terrorist broadcast the murder of more than 50 people on Facebook live stream. After that incident, Facebook shared this information with other platforms. Every video or image uploaded by ordinary users from any of these platforms would now be

[1] https://pypi.org/project/pyleetspeak/.
[2] ar, az, da, de, el, en, es, fi, fr, hu, id, it, kk, nb, ne, nl, pt, ro, ru, sl, sv, tg, tr.
[3] https://github.com/Huertas97/XX_NER_WordCamouflage.
[4] https://huggingface.co/Huertas97/xx_LeetSpeakNER_mstsb_mpnet.

Table 1
Examples of Leetspeak technique applied in different situations, as shown in real-world cases documented by previous studies and references. These examples illustrate the diverse and creative ways in which malicious actors may modify text using Leetspeak to evade content moderation.

Case of study	Original	Camouflaged	Source
Gaming	noobs owned skiils fear	n00bz pwn3d sk11lz ph34r	[25]
Password	HBOpassword	#B0p4$$w0r|)	Hong et al. 2021
Cybersquatting	incibe.es	incive.es incIbe.es inci-be.es inicbe.es	Instituto Nacional de Ciberseguridad (INCIBE)
Social Media COVID-19 Infodemic	vacuna	v4cun4 b4cun4 v@(u¬a nacuva V.A.C.U.N.A	EU Disinfolab
	covid	k0 b1t K0b1d c0*vid C(o(v(i(d	

checked against it to check whether it should be blocked or not [18,23].

In the literature, many examples illustrate the significant role that automated systems are already playing in content moderation. According to [26], 98% of the videos YouTube removes for violent extremism are flagged by machine learning algorithms and help human reviewers remove videos nearly five times faster. Moreover, the visualization time for harmful antivaccine content videos has been reduced by more than 70% after YouTube limited the recommendation for such videos [27]. In 2021, Twitter temporarily locked Donald Trump's account for allegedly inciting an attack on the United States Capitol and permanently suspended it for violating Twitter's Glorification of Violence guidelines [28]. Twitter has also been testing features to allow people to report potentially misleading information, and has recently expanded to Brazil, the Philippines, and Spain [29]. Additionally, in 2021, Facebook has started a campaign to explicitly ban "any content that denies or distorts the Holocaust" [30]. Similarly, other researchers have also developed machine learning solutions to help security administrators detect phishing content in emails and social networks [31]. Other researchers [32,33] have explored the application of machine learning classifier on online social networks to facilitate content-based filtering assistance to avoid unwanted content displayed on the user wall.

These situations and incidents like those raised during the COVID-19 pandemic clearly show that automated moderation systems have become necessary to manage growing public expectations for greater responsibility, safety, and security on the platforms [14,34]. Nevertheless, content filtering automated systems depend on their ability to analyze the material uploaded, potentially vulnerable to recent content evasion techniques, such as word camouflaging.

2.2. Content moderation evasion techniques

Before proceeding further, it is essential to provide a sensitive content disclaimer to inform readers that there is a possibility of encountering offensive content in the examples included within this work.

Community-driven Internet spaces, especially social networks, have always presented unique dialects and slang terminology [25,35]. These ever-changing dialects are the natural result of short codes to facilitate communication, user interactions on social networking platforms, and the adaptation of language to new technologies [15]. However, the emergence of new terms or dialects can result from intentionally camouflaging messages without impacting the information transmitted to avoid content moderation.

One of the main techniques for this purpose is *leetspeak*. *Leetspeak* is a written language in which characters are changed to other characters or combinations of characters that visually resemble the original [25,36] (see Table 1). Even though there is still a great deal of uncertainty about its origin, there is no doubt that its initial use was related to content evasion.

One reliable hypothesis [25] holds that hackers initially used leetspeak in the early stages of the Internet to prevent their content from being accessible. At that time, most search systems searched for keywords in the text to recommend relevant content, and users who were reluctant to share their information substituted certain letters in words to avoid being included in searches. Other observations [37] indicate that leetspeak' origins can be found on Bulletin Board Systems, much like today's forums, to avoid censorship measures present in instant messaging systems.

There is also controversy as to why the term leetspeak was coined. In their analysis of the adaptation of language by a community of young people who play computer games, Blashki et al. [25] proposed that the root term "leet" was originated from 31 337 "eleet", the UDP port used by a hacker group to access Windows 95 using the Back Orifice hacking program. Other researches [37] propose that it was first considered "elite" as only a few people could encode and decode it, therefore using the term "leet" to refer to this particular group. Subsequently, the use of leetspeak became more popular, and it became integrated into the gaming community, particularly by Counter Strike and World of Warcraft players [25,37].

Nowadays, leetspeak is mainly associated with online multiplayer gamers [35]. Interestingly, the leetspeak camouflaging technique is also explored in the field of password generation and password security. Golla et al. [38] mention the practice of using leetspeak to modify characters in passwords to make them more secure, but at the same time facilitate remembering them [36]. Passwords replaced with leetspeak have been tested with password strength estimators, such as *zxcvbn* [39], and get a high-security rating, as they combine alphabet characters, numbers, and special symbols [40]. In the same way, leetspeak has been related to cybersquatting, the registration of a domain name that is the trademark of another [41] (see Table 1).

Remarkably, the work of Peng et al. [42] reveals the vulnerability of machine learning algorithms in detecting spam in emails when they included leetspeak. Because leetspeak uses unconventional spelling and punctuation, the revealed the difficulty for the automatic systems to accurately identify and interpret the words and phrases being used. Hence, this can make it easier for attackers to evade detection and spread harmful or illegal content. To address this vulnerability, in the present work, Transformer-based models are developed and trained on a wide range of text inputs, encompassing examples of word camouflage, with the aim of enhancing their accuracy and ability to identify such language.

Previous studies [43,44] have analyzed a variety of slangs, including leetspeak, in social media, but since the emergence of coronavirus in 2019, this situation has become more pronounced, with leetspeak being a way to circumvent censorship. In [16] the authors analyzed how malicious actors camouflage virus-related Spanish keywords to spread misleading content, revealing their skills in developing new techniques to continue spreading their message and the complexity of tackling this phenomenon (see Table 1).

Finally, evidence of the presence of these methods of content evasion in social networks can be clearly observed through a comparison of the search results for terms associated with hateful behavior and which do not comply with the Community Guidelines in original and camouflaged formats.

The term "self-harm" on the TikTok platform is associated with the health and well-being of people and is part of their community guidelines for safety. Searching for this term redirects to an official contact for hope support.[5] Nevertheless, by altering the search term to "s3lf-harm" moderation is bypassed, granting access to videos with potential content that may depict, promote, normalize, or glorify activities leading to suicide or self-harm.[6] Other examples are the term "incel", which in an extreme way refers to those who have violent or hateful attitudes towards women or towards society in general [45]. This term does not return any results and redirects us to the community guidelines.[7] Once more, moderation can be bypassed and content accessed by employing "1ncel" as an alternative.[8] The same happens in other languages. For example, the search for the Spanish terms "violación" and "pornografía", in English "rape" and "pornography", does not return any results,[9][10] unlike their camouflaged version "v1olaci0n"[11] and "p0rnograf!a"[12] with content with more than 8 million views.

As a result of these findings, developing tools that mimic and detect content avoidance techniques are essential for content moderation in the fight against information disorders.

3. Methodology

This section presents the methodology for generating camouflage data, which aims to address RQ1 by designing a novel, customizable methodology for simulating multilingual content evasion through word camouflage.

First, Section 3.1 outlines the simulation of camouflage techniques using the customizable public Python package developed to apply specific text transformations. Based on literature Refs. [15,16,25,37,46] and strategies observed on social media, three classes of camouflage modifications are proposed: Leetspeak, Punctuation, and Inversion. These techniques will be discussed in further detail in the following subsections.

Secondly, in Section 3.2 the application of these camouflaging techniques to an input text is explained in order to obtain a camouflaged version of the input text with NER annotation, which serves as training and evaluation data for the models.

3.1. Simulation of word camouflage techniques

Taking into account the importance of linguistic characters used in a language to generate camouflaged versions, it is necessary to point out that the tools developed in the package are multilingual (+20 languages[3]). This tool has been tested in English, Spanish, French, Italian, and German. However, it can be easily extensible to the rest of Latin-derived alphabet languages (i.e., most of the Western European languages). As a reminder, the tool described below is publicly available in the Python "pyleetspeak" package[2].

Three approaches have been designed to emulate content evasion strategies based on text camouflage modification. These methods were developed in relation to the results described in the analysis by Romero-Vicente et al. [16] of recent content avoidance techniques on social networks: LeetSpeaker, PunctuationCamouflage and InversionCamouflage modules.

3.1.1. LeetSpeaker module

This module applies the well-known *leetspeak* method to produce visually similar character strings by replacing alphabet characters with special symbols or numbers. There are many ways to use leetspeak, from basic vowel substitutions to advanced combinations of various punctuation marks and symbols.

The *leetspeak* alphabet in its simplest form substitutes vowels, but it can be pretty complex when substituting consonants as well. As a consequence, the leetspeak modifications implemented in the package are organized into five different modes depending on the visual complexity of the camouflage. The implemented changes, available in our repository,[13] have been obtained from different sources [15,16,25,36,37,39,46]. Nevertheless, the tool is flexible, as new possible substitutions can be specified for its adaptability to new unexplored or ever-evolving scenarios.

Similarly, in the case of LeetSpeaker, other parameters that can be set to customize the camouflage result are the probability of changing a character type (e.g., change "a" for "@") and the frequency of substitution, named *chg_prb* and *chg_frq* in the package, respectively. The frequency of substitution refers to the number of positions to change among all matches of the same character (e.g., whether to replace the two "a" letters with the "@" symbol in "vaccination"). In the case of an original character that has more than one possible substitution, one is randomly selected from a uniform probability distribution.

In a real scenario, such as social networks, users tend to use the same type of substitution for all occurrences of the same character. To emulate this situation, it is possible to define whether or not the transformations performed by LeetSpeaker and PunctuationCamouflage should be independent of each other using the *uniform_change* parameter. In other words, it determines whether the same substitution character should be used in all positions where the original text is modified. For example, if "a" can be replaced by "@" or "4", select whether "vaccination" should be "v4ccin4tion" or "v@ccin4tion".

3.1.2. PunctuationCamouflage module

Another method to create visually similar character strings is to insert punctuation symbols into the text (see Table 2).

[5] https://www.tiktok.com/search?q=self-harm.
[6] https://www.tiktok.com/search?q=s3lf-harm.
[7] https://www.tiktok.com/search?q=incel.
[8] https://www.tiktok.com/search?q=1ncel.
[9] https://www.tiktok.com/tag/violacion.
[10] https://www.tiktok.com/search?q=pornografÃ±a.
[11] https://www.tiktok.com/tag/v1olaci0n.
[12] https://www.tiktok.com/search?q=p0rnograf!a.
[13] https://github.com/Huertas97/pyleetspeak/blob/main/pyleetspeak/modes.py.

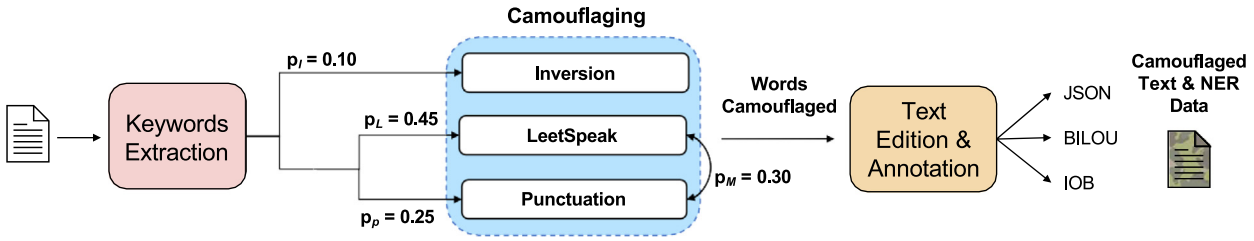


Fig. 1. The diagram illustrates how Named Entity Recognition data is generated using the Python tool "pyleetspeak". The first step in the process is the "Keyword extraction" module. This module uses semantic knowledge transformers to select the terms that will be camouflaged. Camouflaging is then accomplished using leetspeak, inversion, and punctuation camouflage. The probabilities of applying these techniques are represented by p_L, p_I, and p_P, respectively. The probability of applying a mixture of different techniques once a technique has been previously applied is represented by p_M. In the last step, the annotation module annotates the modified terms and positions, resulting in camouflaged data in JSON, BILOU, and IOB formats.

Table 2
Examples of word camouflage using different methods from pyleetspeak.

Word	Leetspeaker			Punctuation	Inversion
	Basic	Intermediate	Advanced		
Vacuna	V@c_n@ V∆cün∆	\/4[u||4 V.qün∆	|/a[L|/\/a \/4[V||/\	'V'a'c'u'n'a Vac'u=na	nacuVa
Covid	C0v1d Cøv¡d	k.vb!t C0▼!t	[ov|d C[]|!|>	?C?o?v?i?d 'C-ovid	vidCo
Plandemia	Plnd£m* Pland.mi∆	₽ la¬t € mla P1∆Π d3ûia	|>landem[]/\ Pl/\nde[V]1a	!P!l!a!n!d!e!m!i!a Plan/demi/a	dePlanmia
Inmigrant	1nmigr_nt Inm*grnt	IΠmig|∆nt Inû¡g| aΠt	In[]V[]igr/\nt In^^][(_+ra||t	.I.n.m.i.g.r.a.n.t +I+nmi+gran	migrantIn
Dictatorship	Dict*t*rship Dict∆t0rship	Dict. ₮ørs #i ₽ Di©tat*rz#lp	Ditat(0)/2ship (|ict/\t<>rship|7	Dicta:torsh:ip D|i|c|t|a|t|o|r|s|h|i| p	Dicortatship
Genocide	Genøcld G%nocide	G3Πo©ite Gen©id@	9e||o[i[)e Gen<>ci|>e	G;enoc;i;de G=e=n=o=c=i=d=e	oGencide

Regarding PunctuationCamouflage module, it can be further customized to inject punctuation symbols in hyphenate locations (i.e., syllables) or between any character. In addition, the number of punctuation symbols to inject can be specified. Interestingly, Romero-Vicente et al. [16] reported that malicious actors usually use punctuation camouflaging, inserting punctuation symbols between all letters of keywords (e.g., "C.O.V.I.D.-1.9"). This behavior can also be reproduced in pyleetspeak without previously specifying the input text length to be camouflaged using the *word_splitting* parameter. The default punctuation symbols applied come from the built-in Python "string" module, but the user can specify the symbols to use.

3.1.3. InversionCamouflage module

Although not as common as the previous methods, word inversion can also be used to confuse moderating algorithms. For this reason, InversionCamouflage module creates new camouflaged versions of words by inverting the order of the syllables (see Table 2).

Word inversion is implemented by detecting the syllables that constitute a word. Once the word has been separated into its syllables, two of these syllables are randomly selected and inverted with respect to each other. As in the previous methods, the inversion is customizable and it is possible to indicate the maximum distance between two syllables to be interchanged. If there are several possibilities for syllable interchange, one is chosen at random.

Further details on the adopted methodology can be found in our repository. Similarly, this tool can be tested in the demo application Leetspeaker App[14] developed with Dash [47], and other examples of the package "pyleetspeak" package are shown in Table 2.

[14] https://github.com/Huertas97/LeetSpeaker_App.

3.2. Word camouflage NER data generator

As depicted in Fig. 1, "pyleetspeak" package transforms an input text into a camouflaged version. The use of word camouflaging usually involves changing the most critical words of a sentence instead of leetspeak all the words in the text. Thus, KeyBERT [48] is used to extract the most semantically relevant words and apply them different word camouflaging methods presented above. Finally, the camouflaged entities in the output text are annotated in Spacy format [49].

KeyBERT incorporates state-of-the-art Transformer models for keyword extraction [12]. This method represents a viable alternative to traditional statistical methods for keyword extraction, as it benefits from the use of powerful Transformer-based models [50,51]. These state-of-the-art models have recently radically transformed the Natural Language Processing area for their ability to generate powerful semantically aware text representations. Precisely, KeyBERT exploits this semantic awareness to compute words and text embeddings, then extracts the most semantically relevant words of a text using cosine similarity as similarity function. Consequently, the most similar words are the keywords that best describe the meaning of the text. By incorporating KeyBERT into the NER data generator, the expectations of real-world scenarios are better met, where malicious actors camouflage crucial concepts in conversation to evade content moderation. Additionally, the model *mstsb-paraphrase-multilingual-mpnet-base-v2* fine-tuned on the multilingual Semantic Textual Similarity Benchmark [17] is employed as the Transformer model for keyword extraction. However, any other HuggingFace [52] model can be selected. The tool has been optionally adapted to always incorporate user-specified keywords to better control the camouflaged output.

Finally, the camouflaged NER annotated data generated is composed of 4 different types of entities. LEETSPEAK,

PUNCT_CAMO, INV_CAMO represent the different camouflage methods implemented, and the MIX entity, which represents the combination of leetspeak and punctuation camouflage. It is also worth noting that, in order to increase the interpretability of the process, besides the annotated camouflage data, the tool returns a dictionary containing the parameters applied to each instance (e.g., keywords extracted, type of camouflage applied, values of the parameters).

4. Experimental setup

This section presents the collected multilingual non-camouflaged text data, which is then camouflaged using the methodology introduced in the previous section to simulate content evasion and address RQ1 by creating a curated synthetic multilingual dataset to support research in content evasion detection. The resulting dataset serves as the foundation for addressing RQ2 and RQ3 by training monolingual and multilingual models for NER word camouflage detection. Additionally, this section provides an overview of the considered models and their configurations for the experiments.

4.1. Non-camouflaged training data

To the best of our knowledge, no dataset with annotated word camouflage modifications is available to train and evaluate our models. Although there are some existing datasets in the field, none of them meet our specific needs and, therefore, cannot be utilized directly. In fact, one of the main contributions of this work is the creation of a new publicly available dataset[4] which fills the existing gap and is anticipated to be beneficial for other researchers in the future.

The dataset with synthetic camouflaged words is elaborated from non-camouflaged texts since, to train the models for word camouflage detection, the camouflage modifications present in the text must be previously annotated. Hence, non-camouflaged datasets are employed, as it ensures that the camouflaging originates exclusively from the modifications derived by the word camouflage generator tool.

The following resources have been chosen due to the variety of text types they encompass:

- **OPUS News-Commentary** [53]: A parallel corpus of political and economic news commentaries in 12 languages was crawled from the web site Project Syndicate provided by WMT.
- **OPUS ParaCrawl** [53]: Multilingual parallel corpora from around 150k website domains and across 23 EU languages collected in the ParaCrawl project [54] cofinanced by the European Union.
- **TED2020** [55]: This dataset contains a crawl of nearly 4000 TED and TED-X transcripts from July 2020. The transcripts have been translated by a global community of volunteers into more than 100 languages.
- **WikiMatrix** [56]: Mined parallel sentences from the content of Wikipedia articles in 85 languages. In this project, a 1.04 score threshold was used for parallel text extraction.

The languages considered in this work are English, Spanish, French, Italian, and German. For each language, data is extracted from the different resources shown above, discarding those texts with a length of less than 3 characters. Subsequently, the extracted data are camouflaged, discarding the annotated data that do not pass the Spacy quality filter[15] and are split in a stratified way into train (81%), validation (9%) and test (10%) sets to perform training and evaluation of multilingual and monolingual NER models. The breakdown of the final data considered in this work according to the language and type of resource can be found in Table 3 and is publicly available on GitHub[4].

The parameters used to camouflage the data with the camouflage tool previously presented and how the quality of the data is evaluated with the Spacy tool will be explained in Section 4.2 below.

4.2. Annotated NER data: Camouflaging parameters and quality filter

To carry out camouflage and annotation of the modified words in the data presented in the previous section, different occurrence probabilities were assigned to the camouflage methods using the methodology presented in Section 3. As depicted in Fig. 1, word inversion is utilized in 10% of the cases, whereas leetspeak is applied with a probability of 45%, punctuation camouflage with 25%, and a combination of both with 30%. Although the use of inversion modification in conjunction with other camouflage techniques is potentially feasible, no evidence has been found to support this approach. Therefore, in this work, inversion modification has been applied in a stand-alone mode, and the camouflaging techniques are distributed in a way that best reflects reality, though they are fully customizable.

For the sake of reproducibility, all the values used in the various parameters to generate the NER data are shown in Table 4. To ensure that the different train, validation and test splits have the same distribution of entity types, the parameters of the NER data generator are equally employed across languages to create the camouflaged data version.

Upon obtaining the modified datasets, the Spacy data debugger[16] is applied. This quality filter removes possible duplicates from the source data, checks that there are no overlaps between the training and evaluation data, that there are a good number of examples for all labels, there are examples with no occurrences available for all labels, there are no entities consisting of or starting/ending with blanks, and there are no entities crossing sentence boundaries.

Finally, the camouflaged annotated data obtained are saved in Spacy formats. The customizable "pyleetspeak" tool presented also provides a format converter to transform Spacy NER format to JSON, BILUO, or IOB format, the one used by the Hugging Face community.

The curated synthetic multilingual dataset obtained is available on GitHub4 in the different formats indicated above.

4.3. Word camouflage NER models

As mentioned above, one of the objectives of this work is to develop a multilingual detector model to address the problem of content evasion by word camouflage from a multilingual perspective. Likewise, another objective is to continue the research carried out in our previous work [17] focused on the usefulness of semantic similarity as a generalization task at the multilingual level. This is why the model developed in the previous work and its baseline, as well as other notable multilingual models, have been included. Consequently, the multilingual models that have been adapted for the Named Entity Recognition task of camouflaged words are presented below. In the subsequent sections of the article, the various multilingual models tested are referred to by their abbreviations. This is done to make the text more concise and readable, while still providing sufficient information for the reader to comprehend the context.

15 https://spacy.io/api/cli#debug-data.

Table 3
Breakdown of the multilingual data corpus after quality filtering using Spacy data debugger[16]. This filter ensures the removal of possible duplicates and checks for overlaps between training and evaluation data. It also ensures a sufficient number of examples for all labels, the absence of entities consisting of or starting/ending with blanks, and the absence of entities crossing sentence boundaries. The table is organized by language, resource, and division (training, development, or test), to develop NER word camouflage models.

	EN			ES			FR			IT			DE		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
News commentary	645	73	80	11 341	1255	1398	589	69	72	130	15	16	859	94	103
ParaCrawl	19 678	2180	2445	23 113	2580	2853	21 424	2373	2656	22 279	2474	2772	20 813	2334	2563
TED2020	15 548	1712	1907	15 717	1758	1938	14 225	1580	1751	15 091	1680	1879	14 742	1667	1818
WikiMatrix	15 557	1707	1927	15 529	1717	1903	14 489	1626	1806	15 124	1662	1889	14 457	1618	1782
Total	51 428	5672	6359	65 700	7310	8092	50 727	5648	6285	52 624	5831	6556	50 871	5713	6266

Table 4
Summary of parameter values used to camouflage and annotate text data to develop NER models.

LeetSpeaker	change_prb = 0.8 change_frq = 0.5 probability basic modes = 0.5 probability intermediate modes = 0.4 probability advanced mode = 0.1 probability uniform_change = 0.6
Punctuation Camouflage	probability hyphenation = 0.5 probability uniform_change = 0.6 probability word_spliting = 0.5 number injections = randint(1, lenght)
Inversion Camouflage	max distance = randint(1, 4)
KeyBERT	model = mstsb-paraphrase-multilingual-mpnet-base-v2 max number of keywords = 5 keywords n_gram = (1, 1)

- **paraphrase-multilingual-mpnet-base-v2 (MPNET-base)**: Distilled version of the MPNet model from Microsoft [57] fine-tuned with large-scale paraphrase data using XLM-RoBERTa as the student model. This model is included as a baseline to corroborate the usefulness of the multilingual pre-train in semantic similarity showed in our previous work [17], as it does not includes any fine-tuning on semantic similarity.
- **mstsb-paraphrase-multilingual-mpnet-base-v2 (MPNET-ideal)**: Previous model fitted with multilingual train data from the Semantic Textual Similarity Benchmark (STSb) [58] extended version to 15 languages (mSTSb) [17]. This model has shown to enhance the performance across languages, outperform monolingual models and the capability of generalize to new tasks. This model has been presented previously in the 22nd International Conference on Intelligent Data Engineering and Automated Learning (IDEAL) [17].
- **bloomz-560m (BLOOMz)** [59]: Fine-tuned variant of the pre-trained multilingual BLOOM [60] and mT5 [61] model families on cross-lingual task mixture of 13 training tasks in 46 languages with English prompts capable of following human instructions in dozens of languages zero-shot. The version used is the version of 560M parameters.
- **xlm-roberta-base (XLM-R)**: Base-sized XLM-RoBERTa [62] model totalizing ∼125M parameters. XLM-RoBERTa is RoBERTa model [63], robust version of BERT, pre-trained on CommonCrawl data containing 100 languages.
- **Model 5 - bert-base-multilingual-cased (mBERT)**: BERT [64] transformer model pre-trained on a large corpus of 104 languages Wikipedia articles using the self-supervised masked language modelling (MLM) objective with ∼177M parameters.

The best multilingual model obtained for the NER task is compared with the monolingual model fine-tuned for the NER task in each language included. Therefore, the following monolingual models are considered as baseline:

- **roberta-base** [63]: English baseline model. Pre-trained model in English using a masked language modeling (MLM) on Wikipedia articles and BookCorpus [65].
- **roberta-base-bne** [66]: Spanish baseline model. Masked language model for the Spanish language based on the RoBERTa base model pre-trained using spanish web crawlings performed by the National Library of Spain from 2009 to 2019.
- **camembert-base** [67]: French baseline model. Masked language model for French based on the RoBERTa base model pre-trained using the French portion of the Open Super-large Crawled Aggregated coRpus (OSCAR) [68].
- **robit-roberta-base-it** [69]: Italian baseline model. Masked language model for French based on the RoBERTa base model pre-trained solely on the Italian portion of the OSCAR dataset.
- **gottbert-base** [69]: German baseline model. Masked language model for French based on the BERT base model pre-trained solely on the German portion of the OSCAR dataset.

The models were fine-tuned using the Spacy interface [49] as the camouflage NER data is in Spacy format. For the sake of reproducibility, the parameters and hyperparameters used during the training process can be consulted in Table 5. A more detailed view of these parameters and training metrics is available at Weight & Biases.[16] Additionally, the models are publicly available on Hugging Face[5] either for direct use or for integration into other Spacy pipelines.

5. Experiments and results

The experiments presented in this section aim to develop the best multilingual NER model for word camouflage detection. Then compare this best multilingual model against the monolingual baseline models. Finally, the detection performance of different camouflage entities is analyzed by visualizing the confusion matrices of the best multilingual model. Hence addressing RQ2 and RQ3.

As the NER word camouflaged detection task considered in this work consists of four imbalanced mutually exclusive classes (see Sections 3.2 and 4.2), the F1 score metric is reported with its different variants, micro, macro and weighted averages. After all, F1 score variants include both precision and recall because they rely on the model's True Positives (TP), False Positives (FP) and False Negatives (FN).

The macro-averaged F1 score represents the unweighted mean; this is computing the arithmetic mean of all the per-class F1 scores. On the other hand, the micro-averaged F1 score

16 https://wandb.ai/aida-group/ASOC-LeetSpeakNER-full-XX-MultiNER/overview.

Table 5
A description of the parameters considered during the training of the models for word camouflaged Named Entity Recognition with Spacy.

learning rate	initial_rate = 0.00005 total_steps = 20 000 scheduler = warmup_linear warmup_steps = 250
epochs	max_epochs = 0 max_steps = 20 000 patience = 1600
accumulate_gradient	3
optimizer	AdamW beta = 10.9 beta2 = 0.999 eps = 1e−8 grad_clip = 1 l2 = 0.01 l2_is_weight_decay = true
eval_frequency	200
dropout	0.1

Table 6
Comparison of the F1-macro, F1-micro, and F1 weighted average test results for the multilingual models considered in the overall test, which includes multiple languages and resources. Model acronyms are the same as those reported in Section 4.3. Italics indicate the second best result, bold indicates the best.

		MPNET-base	MPNET-ideal	BLOOMz	XLM-R	mBERT
News comentary	F1-macro	0.9398	0.9455	0.8221	**0.947**	*0.9458*
	F1-micro	0.9892	*0.9908*	0.9653	**0.9916**	0.99
	F1-weighted	0.8855	**0.9019**	0.6845	0.8843	0.8915
ParaCrawl	F1-macro	0.9306	*0.9316*	0.7942	**0.9336**	0.9274
	F1-micro	0.988	*0.9887*	0.9617	**0.989**	0.9846
	F1-weighted	0.872	**0.876**	0.6457	0.8633	0.8643
TED2020	F1-macro	0.94	**0.9437**	0.8062	0.9397	*0.9404*
	F1-micro	0.988	**0.9893**	0.9597	*0.9883*	0.9867
	F1-weighted	0.8863	**0.8933**	0.6652	0.8746	*0.8837*
WikiMatrix	F1-macro	0.9195	**0.9291**	0.786	0.9239	0.9279
	F1-micro	0.981	**0.9839**	0.9465	*0.9827*	0.9816
	F1-weighted	0.8516	**0.8664**	0.6324	0.8427	*0.8598*
Overall	F1-macro	0.9308	**0.935**	0.7971	*0.9334*	0.9321
	F1-micro	0.9866	**0.988**	0.9582	*0.9876*	0.9849
	F1-weighted	*0.8712*	**0.8795**	0.6499	0.862	0.8698

computes the proportion of correctly classified observations out of all observations, as it computes a global average F1 score by summing the respective TP, FP, FN values across all classes. Finally, the F1-weighted average is calculated by taking the mean of all per-class F1 scores while considering each class's support.

Since the models are tested on an imbalanced dataset, the F1 macro and F1 weighted averages are preferred. It is important to note that the F1-macro metric allows us to evaluate the models considering that all classes are equally important. At the same time, F1-weighted assigns higher contributions to the classes with more examples in the dataset. As explained in the Methodology Section 4.2, not all types of camouflage are applied in the same proportion; hence, particular attention has been given to the F1-weighted mean. Similarly, it should be noted that the test results are reported both generally and broken down by dataset. The overall result is obtained by considering all instances of the datasets as a whole and not the average of the individual results.

5.1. Multilingual NER word camouflage models

The results of NER word camouflage detection in the test partitions for the different trained multilingual models are presented in Table 6.

From these results, it can be observed how MPNET-ideal (see Section 4.3), the one previously developed using the semantic textual similarity pre-training task with the multilingual extended mSTSb dataset [17], shows the best performance in most datasets and is the best in general. It should be noted that MPNET-ideal outperforms MPNET-base, which corresponds to the same model and architecture, but without pre-training in mSTSb. This result corroborates the suitability of the semantic similarity task as a method of providing generality knowledge to a model and the benefit of the multilingual extension of a dataset shown in our previous work [17].

Remarkably, the models perform adequately in the different scenarios, since there are no significant differences in the results obtained between the other datasets. However, the WikiMatrix and ParaCrawl datasets show the lowest scores. This could be due to the diversity of symbols present in the data, since both are the result of multilingual Wikipedia and Internet crawls, which can differ between languages and also include URLs, programming code, or mathematical formulas with the variability of symbols that this implies. On the contrary, the News Commentary and TED2020 datasets show better scores since they correspond to natural text in a formal and informal style, respectively, but with less variability of symbols. It should also be noted that good results are obtained in a few shot scenario, such as the News Commentary dataset, which is the dataset with the lowest number of instances (see Table 3). This indicates generalization and transfer knowledge capabilities to different scenarios.

5.2. Monolingual baseline NER word camouflage models

The best multilingual model, MPNET-ideal, is compared with the monolingual baseline models. The results of these models and their comparison are shown in Table 7.

In particular, the test results in the overall dataset show how the multilingual model outperforms the monolingual baseline models. The most significant difference is found in the Italian language. The Italian monolingual baseline model has the lowest weighted F1 score of 0.7061 across languages; however, the multilingual model improves the score to 0.8913. Similarly, in the case of English and French, the baseline performance with an F1 score weighted of 0.7831 and 0.8572 is improved to 0.8126 and 0.8739, respectively. Although not to the same extent, the Spanish and German languages also improve the baseline model scores.

Our experiments are consistent with previous result [17] as corroborate the usefulness of extending a dataset at the multilingual level to improve the performance of a multilingual model over those of monolingual models. This is an advantage in terms of using multilingual models over monolingual ones. Lower computational cost is required, and feasibility and applicability are increased since instead of a monolingual model to detect camouflage in each language, a single model can be used for all of them with better and more consistent results across languages.

5.3. Multilingual NER confusion matrix analysis

In this subsection, the model's capacity to detect word camouflage and distinguish the type of techniques is analyzed in greater detail, by providing the corresponding confusion matrix in Fig. 2. A confusion matrix for each data resource and the overall resource is shown in this figure. It is important to note that the entity "O' comes from "Outside" and refers to those terms that are not entities to be detected.

These confusion matrix visualize the performance of the model and provide important insights into its capability to accurately detect each entity. As expected, one aspect of interest that emerged from the confusion matrices is that, across all datasets,

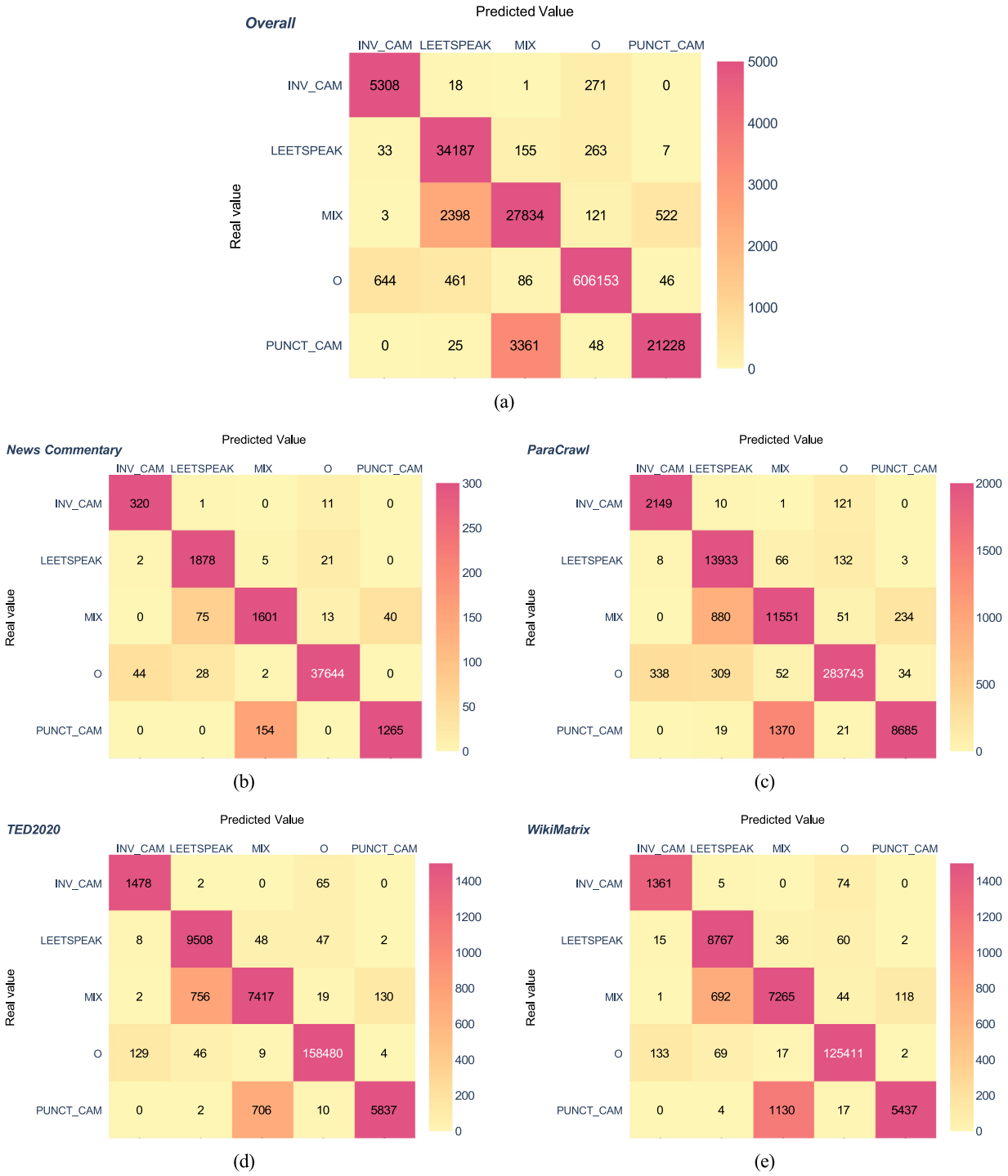


Fig. 2. Entity-level confusion matrix of word camouflage for the best multilingual NER model. (a) The matrix is based on the whole multilingual test data. The results are also broken down by dataset source: (b) News Commentary, (c) ParaCrawl, (d) TED2020, and (e) WikiMatrix. The rows represent the actual camouflage type, and the columns represent the predicted camouflage type by the model. The number in each cell indicates the number of entities. The matrix demonstrates the performance of the multilingual model in detecting different types of word camouflage across various datasets, providing insight into the model's capabilities in different scenarios.

trying to differentiate "MIX" entities from "LEETSPEAK" or "PUNCT_CAMO" entities is more complex as they are closely related due to the fact that MIX includes elements of both of those techniques.

Additionally, these matrices show that detecting inversion camouflage is more difficult than detecting punctuation or leetspeak camouflage, since inversion displays more false positives and false negatives. This is because inversion camouflage is similar to normal text and is challenging to distinguish from it, while punctuation and leetspeak camouflage are more easily distinguishable from normal text. This makes it more difficult for the NER model to accurately differentiate between them.

Taken together, these results suggest that the developed MPNET-ideal multilingual model accurately detects camouflaged entities across multiple languages, and in different types of text with high precision and recall. This suggests that multilingual Transformer models are a promising approach to detecting word camouflage entities in a wide range of contexts.

Table 7
Comparison of F1-macro, F1-micro, and F1 weighted average test results for the best multilingual model (i.e., MPNET-ideal from Section 4.3) and each monolingual baseline model for the languages considered according to the overall and each data set. Bold values indicate the best result for each language and data set. This table serves to show that the multilingual model outperforms its monolingual counterparts in most cases.

		EN		ES		FR		IT		DE	
News comentary	F1-macro	EN	0.9172	ES	0.9501	FR	0.9365	IT	0.8270	DE	0.9607
	F1-micro	Model	0.9864	Model	0.9901	Model	0.9880	Model	0.9612	Model	0.9909
	F1-weighted		**0.8522**		0.9025		**0.8705**		0.6982		0.9210
	F1-macro	MPNET-ideal	0.9167	MPNET-ideal	0.9487	MPNET-ideal	0.8771	MPNET-ideal	0.8783	MPNET-ideal	0.9733
	F1-micro		0.9868		0.9917		0.9756		0.9835		0.9940
	F1-weighted		0.8515		**0.9081**		0.7872		**0.8133**		**0.9484**
ParaCrawl	F1-macro	EN	0.8381	ES	0.9387	FR	0.9338	IT	0.8265	DE	0.9521
	F1-micro	Model	0.9639	Model	0.9852	Model	0.9863	Model	0.9586	Model	0.9884
	F1-weighted		0.7376		0.8921		0.8713		0.6915		0.9048
	F1-macro	MPNET-ideal	0.8690	MPNET-ideal	0.9422	MPNET-ideal	0.9373	MPNET-ideal	0.9396	MPNET-ideal	0.9504
	F1-micro		0.9809		0.9903		0.9909		0.9897		0.9909
	F1-weighted		**0.7750**		**0.8977**		**0.8847**		**0.8849**		**0.9126**
TED2020	F1-macro	EN	0.9102	ES	0.9552	FR	0.9325	IT	0.8535	DE	0.9577
	F1-micro	Model	0.9806	Model	0.9890	Model	0.9840	Model	0.9612	Model	0.9882
	F1-weighted		0.8286		0.9104		0.8704		0.7353		0.9140
	F1-macro	MPNET-ideal	0.9224	MPNET-ideal	0.9528	MPNET-ideal	0.9376	MPNET-ideal	0.9469	MPNET-ideal	0.9574
	F1-micro		0.9854		0.9909		0.9892		0.9898		0.9911
	F1-weighted		**0.8500**		**0.9153**		**0.8844**		**0.8972**		**0.9181**
WikiMatrix	F1-macro	EN	0.8889	ES	0.9323	FR	0.9024	IT	0.8332	DE	0.9417
	F1-micro	Model	0.9724	Model	0.9805	Model	0.9751	Model	0.9524	Model	0.9813
	F1-weighted		0.7998		0.8720		0.8186		0.6991		**0.8829**
	F1-macro	MPNET-ideal	0.9032	MPNET-ideal	0.9349	MPNET-ideal	0.9198	MPNET-ideal	0.9475	MPNET-ideal	0.9380
	F1-micro		0.9785		0.9849		0.9831		0.9882		0.9852
	F1-weighted		**0.8245**		**0.8805**		**0.8503**		**0.8954**		0.8827
Overall	F1-macro	EN	0.8749	ES	0.9434	FR	0.9254	IT	0.8364	DE	0.9512
	F1-micro	Model	0.9712	Model	0.9863	Model	0.9834	Model	0.9578	Model	0.9869
	F1-weighted		0.7831		0.8938		0.8572		0.7061		0.9020
	F1-macro	MPNET-ideal	0.8958	MPNET-ideal	0.9445	MPNET-ideal	0.9320	MPNET-ideal	0.9439	MPNET-ideal	0.9495
	F1-micro		0.9817		0.9898		0.9886		0.9893		0.9898
	F1-weighted		**0.8126**		**0.9002**		**0.8739**		**0.8913**		**0.9069**

5.4. Illustrating synthetic data and model performance in detecting camouflaged entities

To ensure interpretability and transparency, Fig. 3 displays simulated examples created by the "pyleetspeak" tool, along with the performance of the multilingual NER model designed for detecting camouflaged entities. The synthetic data belong to the synthetic test data produced using the input text parameters outlined previously in Table 4.

In addition, an Appendix has been added to incorporate real-life examples in order to demonstrate the proper simulation of the content evasion used on social networks. Likewise, an application[17] has been developed where the model can be tested.

As can be seen, through the use of selected keywords with varying levels of complexity, the tool can effectively simulate word camouflage commonly encountered on social media. In addition to its effectiveness, the tool is also flexible, allowing users to create their own word camouflage at the desired level of complexity. As a result of these alterations, the tool maintains the meaning of the words and mostly preserves their readability.

Remarkably the inversion camouflage technique, which consists of reversing the order of the letters in a word, can be remarkably difficult for the target audience to read and understand. The inversion technique can be particularly confusing due to the reversal of the letters, making it difficult for readers to decipher and understand the text. Consequently, this technique could be used less frequently and stage-dependent. Users should be aware of this aspect when generating data that mimics the scenario to be simulated.

On the other hand, punctuation and leetspeak techniques are more effective at creating data that is more representative of real-world scenarios as shown in Table 1 and Appendix. Moreover, it is evident that word camouflage complexity varies, with the complexity increasing when multiple techniques are applied simultaneously.

Thus, users should consider the type of data they are generating and incorporate appropriate techniques to achieve the best results, since the tool allows users to incorporate or exclude these techniques accordingly.

Overall, the figure provides a comprehensive overview of how the "pyleetspeak" tool can be applied to simulate word camouflaging. It also provides an overview of the multilingual NER model's ability to detect camouflage and identify different camouflage techniques across a variety of languages. The examples demonstrate the model's effectiveness in a variety of contexts and highlight its relevance to addressing content evasion challenges across a variety of linguistic contexts.

5.5. External validation

5.5.1. AugLy external validation and setup

To provide a comprehensive external validation of the dataset and the multilingual detection model, a third-party library from Meta AI called AugLy [70] was employed. Although AugLy's primary objective is data augmentation for text and not specifically word camouflage, some of its methods resemble the camouflage techniques described in the literature and observed on social networks [15,16,25,37,46]. Examples of such methods include replacing letters with similar Unicode or non-Unicode characters, inserting punctuation, changing the text font, and inverting the text. However, there are several differences between AugLy and the pyleetspeak package.

AugLy applies random modifications to the text, without following the patterns described in the literature where specific

[17] https://huggingface.co/spaces/Huertas97/LeetSpeak-NER.

Fig. 3. Examples of word camouflage detection by the multilingual NER model in (a) English, (b) Spanish, (c) French, (d) Italian, and (e) German. These examples showcase the practical use of our synthetic data generated with the "pyleetspeak" tool, simulating real-world cases of word camouflage. By presenting these visualizations, we aim to demonstrate the model's effectiveness in detecting camouflaged words across multiple languages and highlight its relevance to addressing content evasion challenges.

keywords are camouflaged to evade content moderation. In contrast, pyleetspeak uses semantic Transformer models to modify words with higher semantic relevance. Moreover, AugLy is mainly designed for monolingual (English) data augmentation and is less flexible in allowing users to control the types of changes they want to apply. It also does not incorporate word inversion and does not track the changes made to the original text, making it unsuitable for NER applications as it does not return correctly annotated data.

Despite these limitations, AugLy serves as a valuable reference for external validation due to the lack of a benchmark dataset specifically designed for word camouflage. Using AugLy,

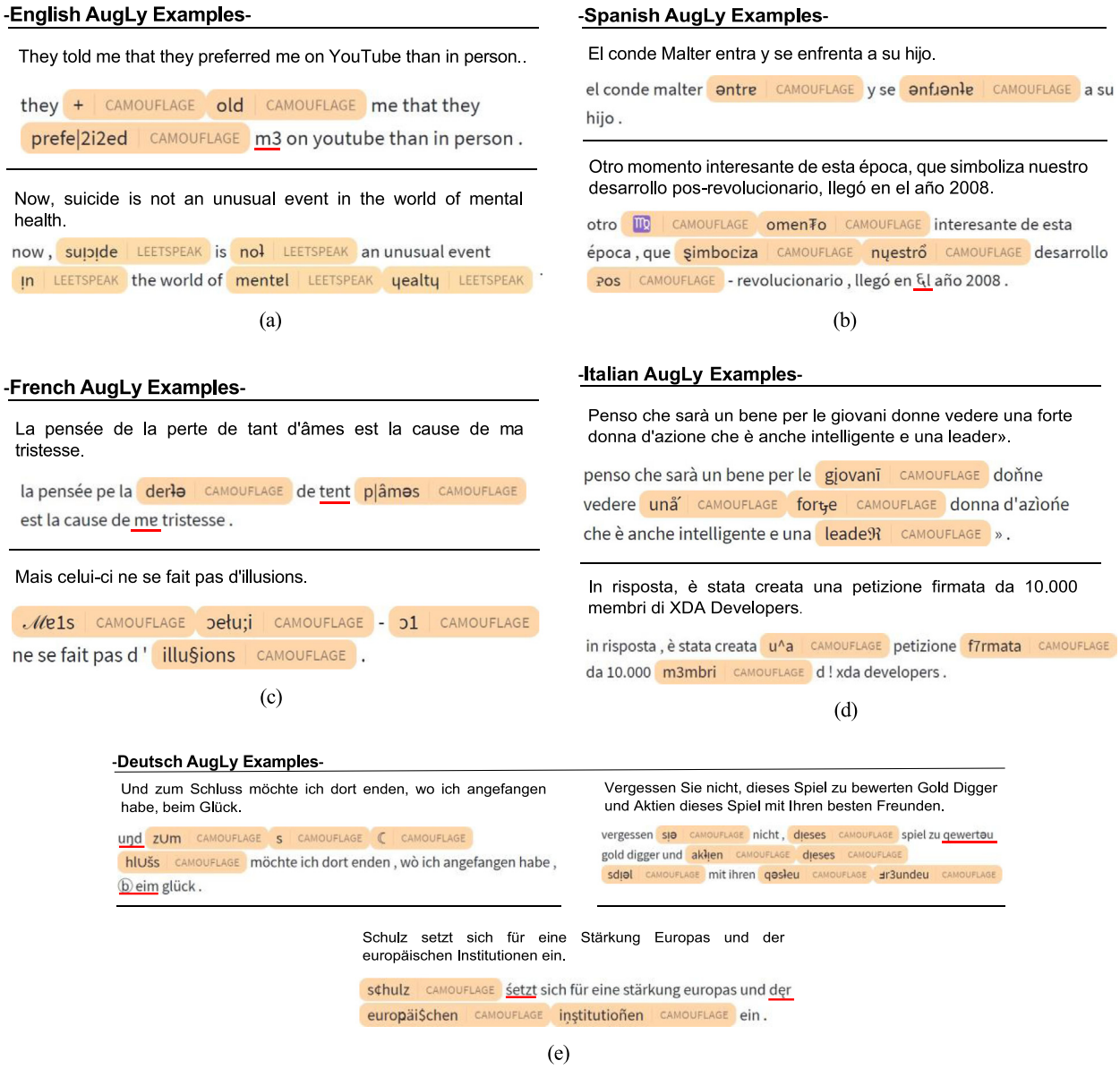


Fig. 4. Examples of word camouflage detection by the multilingual NER model in (a) English, (b) Spanish, (c) French, (d) Italian, and (e) German using AugLy-generated data for external validation. These examples showcase the effectiveness of the model during external validation, even when using data generated by AugLy, which operates differently from the pyleetspeak tool. Camouflaged words that the model fails to detect are underlined in red, often corresponding to non-keywords, illustrating the distinct approaches of pyleetspeak and AugLy in generating synthetic data.

ReplaceSimilarChars, *ReplaceSimilarUnicodeChars*, *ReplaceUpside Down*, *InsertPunctuationChars*, and *ReplaceFunFonts* methods[18] [19] were applied to the original data from each resource and language. The modified data comprised 50% of the dataset, and the task was to evaluate the multilingual model's ability to detect whether a data instance had been modified or no

For data modification, AugLy's default parameters were used, with the exception of granularity, which was set at the word level for all methods except *ReplaceUpsideDown*, which was set at the character level. For the *ReplaceSimi-larUnicodeChars* and *ReplaceFunFonts* methods, the maximum number of words was set to 5, as was the case with the pyleetspeak-generated data, to ensure a comparable validation.

5.5.2. External validation results and insights

As done previously in Fig. 3, to illustrate how the data is transformed by applying AugLy, Fig. 4 shows some examples for the different languages evaluated and the detection performed by MPNET-ideal model.

It can observed that the multilingual model accurately detects unseen novel camouflage strategies such as upside-down letters or emoticons instead of letters, as seen in the first few examples in Spanish and the second example in English. However, the differences between AugLy and the pyleetspeak tool become apparent when transformations are applied to words that are not keywords or do not carry significant semantic weight, such as articles or pronouns. In these cases, the model struggles to detect the modifications, as can be seen underlined in red in the first example in English and the second example in Spanish.

This observation highlights the fundamental difference between AugLy, which inserts transformations randomly throughout the text, and pyleetspeak, which specifically selects the most meaningful words in the sentences for modification. As a

[18] https://augly.readthedocs.io/en/latest/augly.text.html#module-augly.text.

[19] https://github.com/facebookresearch/AugLy/blob/main/augly/text/augmenters/utils.py.

Table 8
Mean F1-macro, F1-micro, and F1-weighted results of the multilingual MPNET-ideal NER model for detecting word camouflage across different languages, based on the external validation with AugLy. Each row represents a language, with F1-macro, F1-micro, and F1-weighted scores calculated as the mean across the different text resources. The table demonstrates the model's effectiveness in identifying camouflage techniques in diverse linguistic contexts, providing insights into its performance, generalizability, and robustness when tested against an independent evaluation source.

Language	F1-macro	F1-micro	F1-weighted
DE	0.7478	0.7622	0.7478
EN	0.9292	0.9296	0.9292
ES	0.8740	0.8762	0.8704
FR	0.9235	0.9240	0.9235
IT	0.8820	0.8836	0.8817
XX	0.8713	0.8751	0.8705

result, the multilingual model may face challenges in detecting modifications in less relevant words like pronouns or articles. This finding underscores the importance of developing camouflage detection models that take into account the diverse nature of content evasion techniques and prioritize the detection of modifications in semantically meaningful words.

The results of the external validation with AugLy in Table 8 highlight the effectiveness of the MPNET-ideal multilingual Named-Entity Recognition (NER) Transformer-based model in detecting a wide range of camouflage techniques, even those not explicitly included during training. The model exhibits stable or improved performance for languages such as English, Spanish, French, and Italian. However, it shows a slightly lower performance for German. This variation may be due to a combination of factors, such as the inherent complexity of German as a morphologically rich language with long compound words and complex grammatical structures, even for non-semantically relevant words, and the possibility that the model learns better representations for other languages sharing more linguistic properties among them. Despite these differences, the overall results demonstrate the model's generalizability and robustness in detecting content evasion across multiple languages on online social platforms, making a significant contribution to combating the spread of malicious information.

6. Conclusions

This study builds on prior research [17] by addressing the growing issue of content evasion in multilingual Natural Language Processing. Content evasion involves altering the wording or formatting of text to dodge detection by automated systems or human moderators, often used to spread misinformation, terrorist, misogynistic, or hateful language content.

This research has successfully addressed the proposed research questions stated in Section 1 by developing the "pyleetspeak" Python package[2], a novel method for simulating word camouflage in social media content with varying complexity tehcniques. The presented methodology have generated a curated synthetic multilingual dataset[4] that maintains the meaning of the original text while altering its form.

These resources have been used to develop a robust multilingual NER camouflage detection model that identifies different word camouflage techniques across more than 20 languages. After comparing different multilingual models, the experiment results confirmed that the model pre-trained on the multilingual Semantic Textual Similarity Benchmark (mSTSb) yielded the best performance outperforming its monolingual counterparts baseline models in five languages: English, Spanish, French, Italian, and German. The proposed tool effectiveness has been demonstrated through experiments and real-life examples in Appendix, showcasing its potential for enhancing content moderation on social media platforms. Additionally, the external validation with AugLy highlights the utility of the dataset and multilingual detection model in addressing the challenging problem of word camouflage in social networks. By providing a comprehensive evaluation, valuable insights into the model's performance and limitations are offered, paving the way for future research and development in this important area.

By addressing the research questions, this study has contributed valuable insights and practical solutions for combating content moderation evasion, ultimately improving online security and reducing the workload of human moderators.

The "pyleetspeak" tool has broader applications, such as a data augmentation tool for enhancing content-dependent AI systems' robustness. Future research will explore word camouflage's impact on real-world scenarios and examine content evasion cases to gain insights into camouflaging methods used in malicious communities.

While tested in five languages, our approach is extensible to other languages and situations, as the tools support over 20 languages with promising results. Future work will also cover different content evasions strategies, such as paralanguage and emoticon use.

CRediT authorship contribution statement

Álvaro Huertas-García: Research, Manuscript preparation, Conceptualization, Data analysis, Methodology, Development, Investigation, Visualization, Writing – review & editing. **Alejandro Martín:** Research, Manuscript preparation, Conceptualization, Data analysis, Methodology, Development, Investigation, Visualization, Writing – review & editing. **Javier Huertas-Tato:** Research, Manuscript preparation, Conceptualization, Data analysis, Methodology, Development, Investigation, Visualization, Writing – review & editing. **David Camacho:** Research, Manuscript preparation, Conceptualization, Data analysis, Methodology, Development, Investigation, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: David Camacho reports financial support was provided by European Commission. Alejandro Martin Garcia reports financial support was provided by BBVA Foundation. David Camacho reports financial support was provided by Spanish Ministry of Science and Education. David Camacho reports financial support was provided by Community of Madrid

Data availability

All links to data and models are provided in the manuscript

Acknowledgments

This research has been supported by the Spanish Ministry of Science and Education under FightDIS (PID2020-117263GB-I00) and XAI-Disinfodemics, Spain (PLEC2021-007681) grants, by Comunidad Autónoma de Madrid, Spain under S2018/TCS-4566 (CYNAMON), by BBVA Foundation, Spain grants for scientific research teams SARS-CoV-2 and COVID-19 under the grant: "*CIVIC: Intelligent characterisation of the veracity of the information related to COVID-19*", and by IBERIFIER (Iberian Digital Media Research and Fact-Checking Hub), funded by the European Commission under the call CEF-TC-2020-2, grant number 2020-EU-IA-0252. Finally, David Camacho has been supported by the Comunidad Autónoma de Madrid, Spain under "Convenio Plurianual with the Universidad Politécnica de Madrid in the actuation line of *Programa de Excelencia para el Profesorado Universitario*"

All authors read and approved the final manuscript.

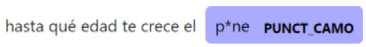
Appendix. Real examples

Link to the resource

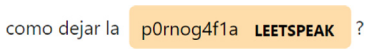
Link to the resource

Link to the resource

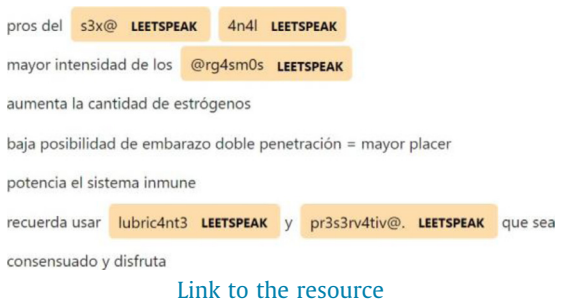
Link to the resource



Link to the resource

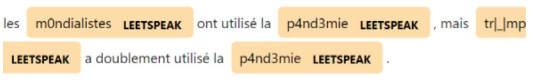


Link to the resource

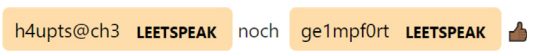


Link to the resource

Link to the resource

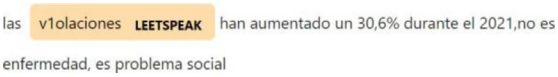


Link to the resource

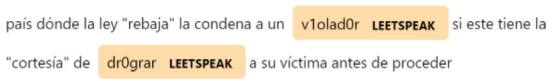


Link to the resource

Link to the resource

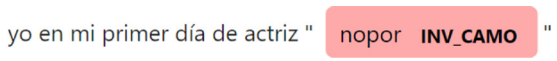


Link to the resource



Link to the resource

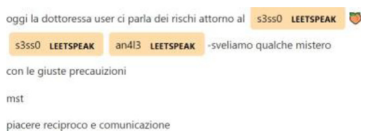
Link to the resource



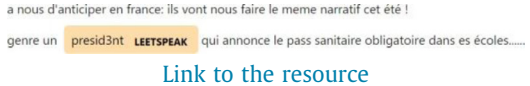
Link to the resource

Link to the resource

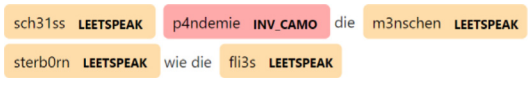
Link to the resource



Link to the resource

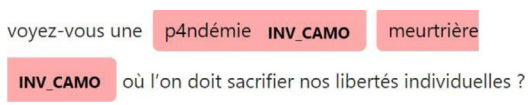


Link to the resource

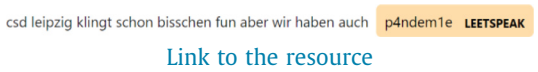


Link to the resource

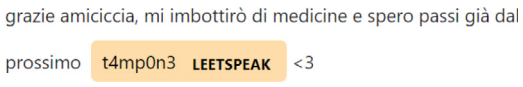
Link to the resource



Link to the resource

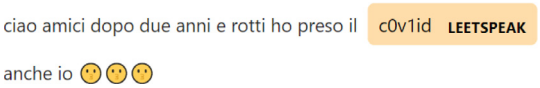


Link to the resource



Link to the resource

Link to the resource



Link to the resource

References

[1] F. Fagan, Optimal social media content moderation and platform immunities, Eur. J. Law Econom. 50 (3, SI) (2020) 437–449, http://dx.doi.org/10.1007/s10657-020-09653-7.

[2] N. Thilagavathi, R. Taarika, Content based filtering in online social network using inference algorithm, in: 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], 2014, pp. 1416–1420, http://dx.doi.org/10.1109/ICCPCT.2014.7054762.

[3] Y. Gerrard, Beyond the hashtag: Circumventing content moderation on social media, New Media Soc. 20 (12) (2018) 4492–4511, http://dx.doi.org/10.1177/1461444818776611.

[4] L. Kelly, G. Kerr, J. Drennan, Avoidance of advertising in social networking sites, J. Interact. Advert. 10 (2) (2010) 16–27, http://dx.doi.org/10.1080/15252019.2010.10722167.

[5] S. Chancellor, J.A. Pater, T. Clear, E. Gilbert, M. De Choudhury, #Thyghgapp: Instagram content moderation and lexical variation in pro-eating disorder communities, in: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing, CSCW '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1201–1213, http://dx.doi.org/10.1145/2818048.2819963.

[6] A. Mosseri, Addressing hoaxes and fake news, 2016, URL https://about.fb.com/news/2016/12/news-feed-fyi-addressing-hoaxes-and-fake-news/.

[7] R. Yoel, N. Pickles, Updating our approach to misleading information. URL https://blog.twitter.com/en_us/topics/product/2020/updating-our-approach-to-misleading-information.

[8] F. Sharevski, R. Alsaadi, P. Jachim, E. Pieroni, Misinformation warnings: Twitter's soft moderation effects on covid-19 vaccine belief echoes, Comput. Secur. 114 (2022) 102577, http://dx.doi.org/10.1016/j.cose.2021.102577.

[9] L.S. Martinez, Health Misinformation and Rumors, John Wiley & Sons, Ltd, 2022, pp. 1–6, http://dx.doi.org/10.1002/9781119678816.iehc0950.

[10] COVID-19 Stream, Twitter Developer Platform, URL https://developer.twitter.com/en/docs/labs/covid19-stream/overview.

[11] Twitter API for Academic Research | Products, Twitter Developer Platform, URL https://developer.twitter.com/en/products/twitter-api/academic-research.

[12] A. Martín, J. Huertas-Tato, Á. Huertas-García, G. Villar-Rodríguez, D. Camacho, FacTeR-check: Semi-automated fact-checking through semantic similarity and natural language inference, Knowl.-Based Syst. 251 (2022) 109265, http://dx.doi.org/10.1016/j.knosys.2022.109265.

[13] Policy on Medical Misinformation About COVID-19, YouTube, URL https://support.google.com/youtube/answer/9891785.

[14] R. Gorwa, R. Binns, C. Katzenbach, Algorithmic content moderation: Technical and political challenges in the automation of platform governance, 2020, http://dx.doi.org/10.31235/osf.io/fj6.

[15] M. Kavanagh, Bridge the generation gap by decoding leetspeak, Inside the Internet 12 (12) (2005) 11.

[16] A. Romero-Vicente, Word camouflage to evade content moderation, 2021, URL https://www.disinfo.eu/publications/word-camouflage-to-evade-content-moderation/.

[17] Á. Huertas-García, J. Huertas-Tato, A. Martín García, D. Camacho, Countering misinformation through semantic- aware multilingual models, in: Intelligent Data Engineering and Automated Learning – IDEAL 2021, Springer International Publishing, 2021, pp. 312–323, http://dx.doi.org/10.1007/978-3-030-91608-4_31.

[18] Y. Gerrard, H. Thornham, Content moderation: Social media's sexist assemblages, New Media Soc. 22 (7, SI) (2020) 1266–1286, http://dx.doi.org/10.1177/1461444820912540.

[19] S. Tabassum, J. Gama, P.J. Azevedo, M. Cordeiro, C. Martins, A. Martins, Social network analytics and visualization: Dynamic topic-based influence analysis in evolving micro-blogs, Expert Syst. (2022) http://dx.doi.org/10.1111/exsy.13195.

[20] C. Lampe, P. Resnick, Slash(dot) and burn: Distributed moderation in a large online conversation space, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, Association for Computing Machinery, New York, NY, USA, 2004, pp. 543–550, http://dx.doi.org/10.1145/985692.985761.

[21] N. Elkin-Koren, Contesting algorithms: Restoring the public interest in content filtering by artificial intelligence, Big Data Soc. 7 (2) (2020) http://dx.doi.org/10.1177/2053951720932296.

[22] J. Cobbe, Algorithmic censorship by social platforms: Power and resistance, Philos. Technol. 34 (4) (2021) 739–766, http://dx.doi.org/10.1007/s13347-020-00429-0.

[23] D. Sumpter, Outnumbered: From Facebook and Google To Fake News and Filter-Bubbles - the Algorithms that Control Our Lives, Bloomsbury Sigma, London, 2018, oCLC: on1035374425.

[24] Ofcom, Use of AI in Online Content Moderation, Cambridge Consultants, 2019, URL https://www.cambridgeconsultants.com/insights/whitepaper/ofcom-use-ai-online-content-moderation.

[25] K. Blashki, S. Nichol, Game geek's goss: linguistic creativity in young males within an online university forum, 2005.

[26] Global internet forum to counter terrorism | about. URL https://perma.cc/44V5-554U.

[27] F. Ferreira, Antivaccine videos slip through YouTube's advertising policies, new study finds, Science (2020) http://dx.doi.org/10.1126/science.abf5402.

[28] Permanent Suspension of @RealDonaldTrump, Twitter Inc, 2021, URL https://blog.twitter.com/en_us/topics/company/2020/suspension.

[29] T. Blog, Nuevo canal para reportar información potencialmente engañosa en Twitter, 2022, URL https://blog.twitter.com/es_es/topics/2022/nuevo-canal-para-reportar-informacion-potencialmente-enganosa-en.

[30] M. Bickert, Removing holocaust denial content, 2020, URL https://about.fb.com/news/2020/10/removing-holocaust-denial-content/.

[31] U. Ozker, O.K. Sahingoz, Content based phishing detection with machine learning, in: 2020 International Conference on Electrical Engineering, ICEE, 2020, pp. 1–6, http://dx.doi.org/10.1109/ICEE49691.2020.9249892.

[32] N. Thilagavathi, R. Taarika, Content based filtering in online social network using inference algorithm, in: 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], 2014, pp. 1416–1420, http://dx.doi.org/10.1109/ICCPCT.2014.7054762.

[33] A.S. Vairagade, R.A. Fadnavis, Automated content based short text classification for filtering undesired posts on facebook, in: 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), 2016, pp. 1–5, http://dx.doi.org/10.1109/STARTUP.2016.7583984.

[34] M. Ghayoomi, M. Mousavian, Deep transfer learning for covid -19 fake news detection in Persian, Expert Syst. 39 (8) (2022) http://dx.doi.org/10.1111/exsy.13008.

[35] A.H. Shaari, K.B.A. Bataineh, Netspeak and a breach of formality: Informalization and fossilization of errors in writing among esl and efl learners, Int. J. Cross-Discip. Subj. Educ. 6 (2015) 2165–2173, http://dx.doi.org/10.20533/IJCDSE.2042.6364.2015.0300.

[36] J. Kavrestad, F. Eriksson, M. Nohlberg, Understanding passwords - a taxonomy of password creation strategies, Inf. Comput. Secur. 27 (3) (2019) 453–467, http://dx.doi.org/10.1108/ICS-06-2018-0077.

[37] J. Fuchs, Gamespeak for N00bs - a Linguistic and Pragmatic Analysis of Gamers' Language (Ph.D. thesis), University of Graz, 2013, URL https://unipub.uni-graz.at/obvugrhs/content/titleinfo/231890?lang=en.

[38] M. Golla, B. Beuscher, M. Duermuth, On the security of cracking-resistant password vaults, in: CCS'16: Proceddings of the 2016 ACM SIGSAC Conference on Computer and Comunication Security, 2016, pp. 1230–1241, http://dx.doi.org/10.1145/2976749.2978416.

[39] D.L. Wheeler, Zxcvbn: Low-budget password strength estimation, in: Proceedings of the 25th USENIX Conference on Security Symposium, SEC'16, USENIX Association, USA, 2016, pp. 157–173.

[40] K.H. Hong, U.G. Kang, B.M. Lee, Enhanced evaluation model of security strength for passwords using integrated Korean and english password dictionaries, Secur. Commun. Netw. 2021 (2021) http://dx.doi.org/10.1155/2021/3122627.

[41] Cybersquatting, Qué es y cómo protegerse, 2019, URL https://www.incibe.es/protege-tu-empresa/blog/cybersquatting-y-protegerse.

[42] W. Peng, L. Huang, J. Jia, E. Ingram, Enhancing the naive bayes spam filter through intelligent text modification detection, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Com- Munications/ 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 2018, pp. 849–854, http://dx.doi.org/10.1109/TrustCom/BigDataSE.2018.00122.

[43] T. Singh, M. Kumari, Role of text pre-processing in twitter sentiment analysis, Procedia Comput. Sci. 89 (2016) 549–554, http://dx.doi.org/10.1016/j.procs.2016.06.095.

[44] Z.Z. Izazi, T.M. Tengku-Sepora, Slangs on social media: Variations among malay language users on Twitter, Pertanika J. Soc. Sci. Humanit. 28 (1) (2020) 17–34.

[45] S. Moskalenko, J.F.-G. González, N. Kates, J. Morton, Incel ideology, radicalization and mental health: A survey study, J. Intell. Confl. Warfare 4 (3) (2022) 1–29, http://dx.doi.org/10.21810/jicw.v4i3.3817.

[46] R. Craenen, Leet speak cheat sheet. URL https://www.gamehouse.com/blog/leet-speak-cheat-sheet/.

[47] P.T. Inc, Collaborative data science, 2015, URL https://plot.ly.

[48] M. Grootendorst, Keybert: Minimal keyword extraction with bert, 2020, http://dx.doi.org/10.5281/zenodo.4461265.

[49] I. Montani, M. Honnibal, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength natural language processing in python, 2020, http://dx.doi.org/10.5281/zenodo.1212303.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017, arXiv:1706.03762.

[51] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019, arXiv:1810.04805.

[52] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T.L. Scao, S. Gugger, M. Drame, Q. Lhoest, A.M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45, http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6.

[53] J. Tiedemann, Parallel data, tools and interfaces in OPUS, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 2214–2218.

[54] M. Bañón, P. Chen, B. Haddow, K. Heafield, H. Hoang, M. Esplà-Gomis, M.L. Forcada, A. Kamran, F. Kirefu, P. Koehn, S. Ortiz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarrías, M. Strelec, B. Thompson, W. Waites, D. Wiggins, J. Zaragoza, ParaCrawl: Web-scale acquisition of parallel corpora, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4555–4567, http://dx.doi.org/10.18653/v1/2020.acl-main.417.

[55] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, 2020, arXiv preprint doi:arXiv:2004.09813.

[56] H. Schwenk, V. Chaudhary, S. Sun, H. Gong, F. Guzmán, Wikimatrix: Mining 135 m parallel sentences in 1620 language pairs from wikipedia, 2019, arXiv:1907.05791.

[57] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, 2020, arXiv:2004.09297.

[58] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, L. Specia, Task 1: Semantic textual similarity multilingual and crosslingual focused evaluation, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vol. 2017, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1–14, http://dx.doi.org/10.18653/v1/S17-2001.

[59] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T.L. Scao, M.S. Bari, S. Shen, Z.-X. Yong, H. Schoelkopf, X. Tang, D. Radev, A.F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, C. Raffel, Crosslingual generalization through multitask finetuning, 2022, http://dx.doi.org/10.48550/ARXIV.2211.01786.

[60] T.L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A.S. Luccioni, F. Yvon, M. Gallé, et al., Bloom: A 176b-parameter open-access multilingual language model, 2022, arXiv preprint doi:arXiv:2211.05100.

[61] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, 2020, http://dx.doi.org/10.48550/ARXIV.2010.11934.

[62] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2019, http://dx.doi.org/10.48550/ARXIV.1911.02116.

[63] Z. Liu, W. Lin, Y. Shi, J. Zhao, A robustly optimized BERT pre-training approach with post-training, in: Chinese Computational Linguistics: 20th China National Conference, CCL 2021, Hohhot, China, August 13–15, 2021, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2021, pp. 471–484, http://dx.doi.org/10.1007/978-3-030-84186-7_31.

[64] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423.

[65] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: The IEEE International Conference on Computer Vision, ICCV, 2015, http://dx.doi.org/10.1109/ICCV.2015.11.

[66] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C.P. Carrino, A. Gonzalez-Agirre, C. Armentano-Oller, C. Rodriguez-Penagos, M. Villegas, Spanish language models, 2021, arXiv:2107.07253.

[67] L. Martin, B. Muller, P.J.O. Suárez, Y. Dupont, L. Romary, É.V. de la Clergerie, D. Seddah, B. Sagot, Camembert: a tasty french language model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, http://dx.doi.org/10.18653/v1/2020.acl-main.645.

[68] P.J.O. Suárez, B. Sagot, L. Romary, Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures, in: Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, Leibniz-Institut für Deutsche Sprache, Mannheim, 2019, pp. 9–16, http://dx.doi.org/10.14618/ids-pub-9021.

[69] R. Scheible, F. Thomczyk, P. Tippmann, V. Jaravine, M. Boeker, Gottbert: a pure german language model, 2020, http://dx.doi.org/10.48550/ARXIV.2012.02110.

[70] Z. Papakipos, J. Bitton, Augly: Data augmentations for robustness, 2022, arXiv:2201.06494.

4.3 Camouflage is all you need: Evaluating and Enhancing Transformer Models Robustness Against Camouflage Adversarial Attacks

In this instance, the preprint is available as the article has not been published via open access. The final accepted version is available in [242].

Huertas-García, Álvaro, et al. «Camouflage Is All You Need: Evaluating and Enhancing Transformer Models Robustness Against Camouflage Adversarial Attacks». IEEE Transactions on Emerging Topics in Computational Intelligence, 2024, pp. 1-13. DOI.org (Crossref), https://doi.org/10.1109/TETCI.2024.3440181.

CAMOUFLAGE IS ALL YOU NEED: EVALUATING AND ENHANCING LANGUAGE MODEL ROBUSTNESS AGAINST CAMOUFLAGE ADVERSARIAL ATTACKS

A PREPRINT

Álvaro Huertas-García
Department of Computer Systems Engineering
Universidad Politécnica de Madrid
Madrid, Spain
alvaro.huertas.garcia@upm.es

Alejandro Martín
Department of Computer Systems Engineering
Universidad Politécnica de Madrid
Madrid, Spain
alejandro.martin@upm.es

Javier Huertas-Tato
Department of Computer Systems Engineering
Universidad Politécnica de Madrid
Madrid, Spain
javier.huertas.tato@upm.es

David Camacho
Department of Computer Systems Engineering
Universidad Politécnica de Madrid
Madrid, Spain
david.camacho@upm.es

February 16, 2024

ABSTRACT

Adversarial attacks represent a substantial challenge in Natural Language Processing (NLP). This study undertakes a systematic exploration of this challenge in two distinct phases: vulnerability evaluation and resilience enhancement of Transformer-based models under adversarial attacks.

In the evaluation phase, we assess the susceptibility of three Transformer configurations—encoder-decoder, encoder-only, and decoder-only setups—to adversarial attacks of escalating complexity across datasets containing offensive language and misinformation. Encoder-only models manifest a performance drop of 14% and 21% in offensive language detection and misinformation detection tasks respectively. Decoder-only models register a 16% decrease in both tasks, while encoder-decoder models exhibit a maximum performance drop of 14% and 26% in the respective tasks.

The resilience-enhancement phase employs adversarial training, integrating pre-camouflaged and dynamically altered data. This approach effectively reduces the performance drop in encoder-only models to an average of 5% in offensive language detection and 2% in misinformation detection tasks. Decoder-only models, occasionally exceeding original performance, limit the performance drop to 7% and 2% in the respective tasks. Encoder-decoder models, albeit not surpassing the original performance, can reduce the drop to an average of 6% and 2% respectively.

These results suggest a trade-off between performance and robustness, albeit not always a strict one, with some models maintaining similar performance while gaining robustness. Our study, which includes the adversarial training techniques used, has been incorporated into an open-source tool that will facilitate future work in generating camouflaged datasets. Although our methodology shows promise, its effectiveness is subject to the specific camouflage technique and nature of data encountered, emphasizing the necessity for continued exploration.

Keywords Natural Language Processing · Robustness · Adversarial attack

1 Introduction

The rapid advancement of Artificial Intelligence (AI) and its increasing ubiquity in various domains have underscored the importance of ensuring the robustness and reliability of machine learning models [1, 2]. A particular area of concern lies in the field of Natural Language Processing (NLP) [3], where Transformer-based language models have proven to be very effective in tasks ranging from sentiment analysis and text classification to question answering [4, 5].

One of the novel and key challenges in NLP relates to adversarial attacks, where subtle modifications are made to input data to fool the models into making incorrect predictions [3, 6]. A relevant example of this concept is the use of word camouflage techniques. For instance, the phrase "Word camouflage" can be subtly altered to “W0rd cam0uflage" or “VV0rd cam0ufl4g3." While these changes are often unnoticeable to a human reader, they can lead a machine learning model to misinterpret or misclassify the input [7]. This raises substantial ethical issues around misinformation dissemination, content evasion, and the potential for AI systems to be exploited for malicious intent [8, 9].

Real-world implications of these attacks are increasingly evident, with numerous instances of adversarial attacks compromising online content moderation systems, leading to the spread of harmful content [10]. Existing methods to counter such adversarial attacks have limitations and often focus on post-attack detection [11, 12], failing to proactively prevent the occurrence of such attacks. Additionally, these methods often struggle with text data which is discrete in nature, making it challenging to apply perturbation methods that were originally designed for continuous data like images [2, 3, 13].

This study introduces a comprehensive methodology to evaluate and strength the resilience of Transformer-based language models to camouflage adversarial attacks. We examine the vulnerability and robustness of distinct Transformer configurations—encoder-decoder, encoder-only, and decoder-only—in two use-cases involving offensive language and false information datasets.

The research adopts a proactive defense strategy of adversarial training, which incorporates camouflaged data into the training phase. This is accomplished either by statically camouflaging the dataset or by dynamically altering it during training. A key contribution is the development of an open-source tool[1] that generates various versions of camouflaged datasets, offering a range of difficulty levels, camouflage techniques, and proportions of camouflaged data.

The evaluation of model vulnerability is based on an unbiased methodology, drawing from significant literature references [14, 15, 16, 17, 18] and using AugLy [19], a data augmentation library, for external validation. This approach helps ensure that our assessment of model weakness accurately reflects real-world content evasion and misclassification techniques.

In addressing the the current gaps in the field [3], our research provides insights into three critical challenges—perceivability (the extent to which adversarial changes are noticeable), transferability (the ability of an attack to be effective across different models), and automation (the ability to generate adversarial examples automatically) of camouflaged adversarial examples.

Preliminary results indicate considerable susceptibility across various Transformer configurations, with performance drops reaching up to 14% in offensive language detection and 26% in misinformation detection tasks, highlighting the salient requirement for robustness augmentation.

In addition to identifying these vulnerabilities, the research explores the enhancement of models against such threats. Employing adversarial training methodologies that amalgamate both pre-camouflaged and dynamically altered data, the study uncovers promising avenues for resilience improvement. However, the effectiveness of the approach is influenced by several variables, including the complexity of the camouflage technique employed and the nature and distribution of data encountered by the model, thus emphasising the urgent need for further research in this domain.

The research paper is organised as follows: Section 2 reviews pertinent literature, illuminating the gaps in existing methodologies that this research endeavours to address. Section 3 provides a detailed outline of the methodology employed to develop word camouflage adversarial attacks, while Section 4 explicates the procedure undertaken for enhancing and evaluating the resilience of Transformer models. Section 5 presents empirical findings from each stage of the study, evidencing the impact of adversarial attacks on naive Transformer models, alongside the efficacy of the adversarial fine-tuning approach implemented. Finally, Section 6 offers an in-depth discussion of the research’s implications, ethical considerations, and limitations, concluding with recommendations for future exploration.

[1] Omitted for anonymity reasons.

2 Background and Related Work

This section will cover the various aspects of adversarial attacks within the field of Natural Language Processing (NLP). Topics discussed will include the definition of adversarial attacks, the taxonomy of attacks, the measurement of perturbations, and the evaluation metrics for attack effectiveness. It will also explore the impact of these attacks on deep learning models in NLP tasks and potential solutions and defenses against these attacks.

2.1 Conceptual Aspects of Adversarial Attacks in NLP

Adversarial attacks constitute a significant challenge to deep learning models, as they introduce minimal, often imperceptible, changes to input data with the aim of triggering incorrect model outputs [3, 13, 20].

Adversarial attacks fall into two main categories: white-box and black-box attacks. In white-box attacks, attackers possess complete access to the model's architecture, parameters, loss functions, activation functions, and input/output data, allowing them to craft more precise and effective attacks [3, 21, 22, 23]. Conversely, black-box attacks operate under limited knowledge, with attackers only having access to the model's inputs and outputs. Despite the constraints, these attacks can still induce incorrect model outputs and are more reflective of real-world scenarios [3, 6].

Perturbation measurements, controlling the magnitude and direction of alterations, and evaluation metrics, like accuracy and F1 score, allow for a standardized comparison of different adversarial attacks' impact on model performance [3, 13, 7, 12].

Although adversarial threats persist, several defenses have been proposed, including adversarial training, defensive distillation, feature squeezing, and input sanitization [20, 24, 25]. However, these defenses remain susceptible to more sophisticated attacks [3].

These challenges are particularly relevant for Transformer models given their broad usage in NLP tasks, making the investigation of their susceptibility to adversarial attacks and potential defenses highly significant.

2.2 Transformers and Their Significance

Transformer models have revolutionized Natural Language Processing (NLP), offering exceptional performance across a variety of tasks and thereby becoming a leading model in the field [5, 26, 4]. Unlike their predecessors, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), Transformers employ attention mechanisms, allowing for simultaneous input processing and resulting in enhanced performance on context-dependent tasks [27, 28, 29, 30].

Transformer models can adopt multiple configurations. Encoder-decoder models are suitable for tasks like machine translation, requiring the generation of output sequences based on input understanding [4, 31]. Encoder-only models, such as BERT, focus on input representations, ideal for tasks like text classification [32]. Decoder-only models, like GPT, facilitate sequence generation based on a given context, useful in text generation or language translation tasks [33, 34].

In the face of adversarial attacks, the complexity of Transformer models presents both challenges and opportunities [3]. While their superior performance can lead to vulnerabilities, their configurational flexibility could provide potential defense strategies, making the understanding of their architecture essential in the context of adversarial attacks.

2.3 Previous Works

The exploration of Deep Neural Networks (DNNs) for text data has lagged behind image data, though recent research has begun to reveal DNN vulnerabilities in text-based tasks [3, 24]. Studies have shown that carefully crafted adversarial text samples can manipulate DNN-based classifiers [3].

Significant work, such as those by Liang et al.[35] and Cheng et al.[21], has been done in the white-box attack domain, demonstrating the feasibility of misleading DNN text classifiers with adversarial samples. In another study, Blohm et al.[36] focused on comparing the robustness of convolutional and recurrent neural networks via a white-box approach. Guo et al.[37] offered a white-box attack method for transformer models, but their approach neglected token insertions and deletions, compromising the naturalness of the adversarial examples.

Black-box adversarial attacks have gained attention recently. Noteworthy work by Maheshwary et al.[38] presented an attack strategy that generated high-quality adversarial examples with a high success rate. The work of Gil et al.[39] distilled white-box attack experience into a neural network to expedite adversarial example generation. Similarly, Li et al. [40] proposed a black-box adversarial attack for named entity recognition tasks, demonstrating the importance of understanding black-box attacks for developing robust models.

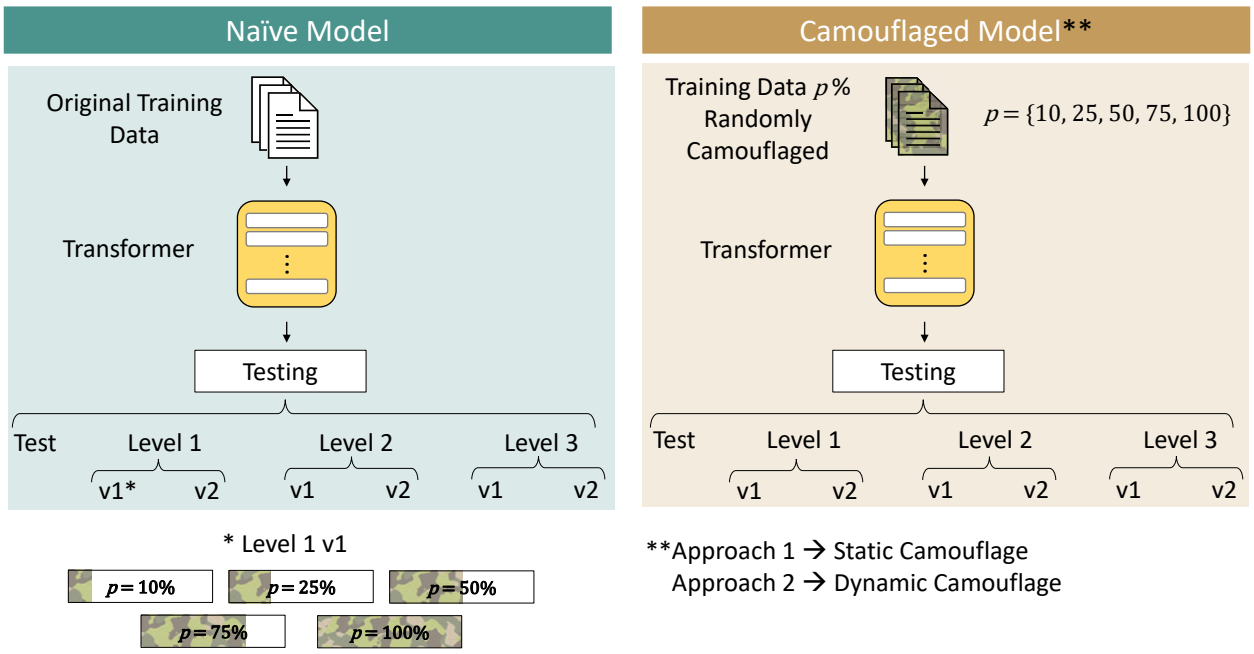


Figure 1: Methodology for training and evaluating Transformed models to assess word camouflage robustness. Left side: naive model trained on original dataset and tested on various versions (Te, C-Te-Lvl1/2/3) with different camouflaged keywords and percentages (p) of modified instances as highlighted in (*). Right side: Camouflaged models trained on data with mixed random level modifications, developed for different percentages (p) of modifications. (**) Two approaches highlighted: Approach 1 with pre-camouflaged training data and Approach 2 with on-the-fly data camouflage during training.

Adversarial attacks have profound real-world implications, ranging from security risks in autonomous driving [41] to societal structures like voter dynamics [42] and social networks [43, 44]. These examples underscore the importance of addressing adversarial attack vulnerabilities across domains, emphasizing naturalness, black-box evaluation, and flexibility.

2.4 Adressing Open Issues

Several challenges persist in the field of adversarial attacks on text data [3]. Primarily, creating textual adversarial examples is complex due to the need to maintain syntax, grammar, and semantics, which makes fooling natural language processing (NLP) systems difficult [2, 3, 13]. Transferability needs a deeper understanding across different architectures and datasets, and automating adversarial example generation remains difficult. Additionally, with the advent of new architectures like generative models and those with attention mechanisms, their vulnerability to adversarial attacks needs to be explored.

To address these issues, the paper proposes an approach focusing on new architectures, transferability, and automation in black-box adversarial attacks. The researchers explore a range of Transformer model configurations, study transferability across different datasets and architectures, and design an automated method for generating adversarial examples. They rely on literature references for unbiased evaluation [14, 15, 16, 17, 18] and AugLy [19] for external validation. The focus on black-box adversarial attacks helps simulate real-world scenarios, contributing to enhanced security in areas like social networks and content moderation.

3 Methodology

This section starts with Phase I, emphasizing the 'Camouflaging Techniques' and 'Camouflage Difficulty Levels' in the development and evaluation of adversarial attacks (subsections 3.1.1 and 3.1.2). Phase II then explores the 'Fine-tuning Approaches' (subsection 3.2.1) to enhance model resilience against word camouflage attacks, followed by an 'External Validation' to ensure robustness (subsection 3.2.2).

Table 1: Parameters defining the levels of word camouflage complexity considered. The table displays three levels of complexity, with each level having two versions based on the max_top_n parameter set to either 5 or 20 for versions 1 and 2, respectively. This parameter defines the maximum number of keywords to extract and camouflage. These levels and versions illustrate the diverse configurations for evaluating the robustness of language models against various camouflage techniques.

	Parameters
Level 1	max_top_n=[5, 20] leet_punt_prb=0.9 leet_change_prb=0.8 leet_change_frq=0.8 leet_uniform_change=0.5 method=["basic_leetspeak"]
Level 2	max_top_n=[5,20] leet_punt_prb=0.9 leet_change_prb=0.5 leet_change_frq=0.8 leet_uniform_change=0.6 punt_hyphenate_prb=0.7 punt_uniform_change_prb=0.95 punt_word_splitting_prb=0.8 method=[ïntermediate_leetspeak "punct_camo"]
Level 3	max_top_n=[5,20] leet_punt_prb=0.4 leet_change_prb=0.5 leet_change_frq=0.8 leet_uniform_change=0.6 punt_hyphenate_prb=0.7 punt_uniform_change_prb=0.95 punt_word_splitting_prb=0.8 inv_max_dist=4 inv_only_max_dist_prb=0.5 method=[ädvanced_leetspeak "punct_camo ïnv_camo"]

Table 2: A description of the parameters considered during the training of the models for word camouflaged Named Entity Recognition with Spacy.

learning rate	initial_rate = 0.00005 total_steps = 20000 scheduler = warmup_linear warmup_steps = 250
epochs	max_epochs = 0 max_steps = 20000 patience = 1600
accumulate_gradient	3
optimizer	AdamW beta = 10.9 beta2 = 0.999 eps = 1e-8 grad_clip = 1 l2 = 0.01 l2_is_weight_decay = true
eval_frequency	200
dropout	0.1

3.1 Phase I: Evaluating the Impact of Word Camouflage on Transformer Models

3.1.1 Camouflaging Techniques

The study employs the “pyleetspeak" Python package, developed based on recent research [45], for generating realistic adversarial text samples. This package applies three distinct text camouflage techniques, prioritizing semantic keyword extraction over random selection, thereby providing a more realistic representation of word camouflage threats. Selected for their relevance and varied complexity in adversarial attacks, these techniques offer a robust evaluation of model resilience. These techniques are:

- Leetspeak: This technique involves substituting alphabet characters with visually analogous symbols or numbers, creating changes that can range from basic to highly intricate. As an illustration, Offensive" could be altered to 0ff3ns1v3" with vowel and specific consonant replacements.
- Punctuation Insertion: This method alters text by introducing punctuation symbols to create visually similar character strings. Punctuation can be inserted at hyphenation points or between any two characters. For example, “fake news" could be camouflaged as “f-a-k-e n-e-w-s".
- Syllable Inversion: This technique, less commonly used, camouflages words by rearranging their syllables. For instance, “Methodology" could be altered to “Me-do-tho-lo-gy" by inverting the syllables in the word.

The use of these techniques helps to perform a thorough investigation of the Transformer models’ resilience to different kinds of adversarial attacks.

	Offen SemEval	Constraint
Original	6 ANTIFA ATTACKED COPS, ARRESTED AT RALLY IN DENVER, MEDIA BLACKOUT URL	FDA chief Hahn comments about convalescent plasma for #COVID-19 patients stirred controversy and a shake up at the agency.
Level 1	6 ANTįFA ATTACKED COPS, ăRRƐST3D AT RALLY IN DēNVėR, MEDĭ∀ BLACKOUT URL	FDą ch!ěf Hahn comments about convalescent plasma for #COVID-19 pàtïƐnts stirred controversy and a shake up at the ∀gency.
Level 2	6 Aи'T1FA ATTACKED :ć:O:P:ś:, ãR]Rҁ५TҁÐ AT RALLY IN |)ƐN;VƐR, mĕÐ!@ BLACKOUT URL	%F%D%A cĥįêf |-|Λh$\mathbb{N}$ comments about convalescent plasma for #CŏVID-19 _|_>_a_t_i_Ɛ_n_t_s stirred controversy and a shake up at the ԁgeиcy
Level 3	6@Д@ŋ@ƥ@į@ſ@Д ATTACKED COPS, ąR>R*e*STƐđAT RALLY IN DēN√ēR, [)1A]V[E BLACKOUT URL	/=(|ä'ch][ə'f Ha'hn c'omme'nts 'abou't co'nval'esce'nt p'lasm'a fo'r #C'OVID'-19 ℙ₳₮ïěnŦs sti'rred' con'trov'ersy' and' a s'hake' up 'at t'he a g ƹ n č y.
AugLy	6 AN...ṫIFA... ATT...ACKE...D CO...PS, ...AŕRE...STEɗ... AT ...RĂLL...Y IN... DEN...VER,... MED...IA B...L@CK...OU7 ...U/2L	FDA chief Hahn ĆommenṮs about convalescent plasma foℜ#COvID-19 patients stiRrĒd controversy and a shake uꝑ aṭ the agency.

Figure 2: Comparison of original and camouflaged text examples from the Offen SemEval 2019 and Constraint datasets. The table presents examples of three levels of camouflage by the tool introduced in this study, as well as an example from the AugLy library for external validation with unseen modifications. Each level represents increasing complexity of camouflage.

3.1.2 Camouflage Difficulty Levels

Adversarial camouflage attacks in this study are systematically varied across three parameters: complexity level, word camouflage ratio (amount of words camouflaged within a data instance), and instance camouflage ratio (the proportion of camouflaged instances within a test dataset). This approach enables a comprehensive assessment of the models' susceptibility to these attacks. Each complexity level involves specific parameters, as detailed in Table 1. Examples of each level are depicted in Table 2.

- **Level 1** serves as a starting point and is quite readable and understandable. It introduces minor changes to the text, primarily focusing on simple character substitutions, such as replacing every vowel with a number or similar symbol.
- **Level 2** increases the complexity of modifications. It introduces extended and complex character substitutions, punctuation injections, and simple word inversions. It extends substitutions to both vowels and consonants, introducing also readable symbols from other alphabets that closely resemble regular alphabet characters.
- **Level 3** is the most complex tier. It combines techniques from the previous two levels but intensifies the use of punctuation marks, introduces more character substitutions, and incorporates inversions, and mathematical symbols are also incorporated making the text even more challenging to comprehend.

For each complexity level, two versions are produced. These versions, known as 'v1' and 'v2', represent different word camouflage ratios, camouflaging 15% and 65% of words per text, respectively. These ratios were derived from the statistical distribution of text lengths in the employed datasets.

These three complexity levels emulate real-world adversarial camouflage attacks, facilitating a methodical progression from low to high complexity, and thus, allowing a detailed evaluation of model robustness against incrementally challenging adversarial scenarios.

Regarding, instance camouflage ratio, as depicted in Figure 1, the tests systematically introduce varying levels and percentages of camouflage into the data instances. Starting from an original test set (Te), 30 additional tests are

generated across three difficulty levels, each with two word camouflage ratio versions (v1 and v2), and five different percentages of camouflaged instances (10%, 25%, 50%, 75%, and 100%). This structure, totalling 31 tests, provides a robust framework for assessing the impact of increasing prevalence and complexity of camouflage techniques. Further discussion on this methodology is covered in Section 4.3.

This systematic approach allows a granular evaluation of word camouflage adversarial attacks, providing insights into their implications and potential countermeasures.

3.2 Phase II: Improving Resilience Against Word Camouflage Attacks

3.2.1 Fine-tuning Approaches

In our study, as depicted in Figure 1, we train and evaluate two distinct sets of models for each task: 'Naive Models' and 'Camouflaged Models'.

The Naive Models represent a baseline approach. These models are trained on the original, unaltered datasets, thus reflecting traditional methods of natural language processing model training. Once training is complete, these models are then evaluated across the 31 different tests we defined earlier, which incorporate varying degrees and proportions of word camouflage.

On the other hand, the Camouflaged Models integrate adversarial data during their training phase. These models are trained on datasets that are modified to include instances of word camouflage, drawn randomly from the three levels of difficulty we previously described. Notably, a distinct Camouflaged Model is trained for each possible percentage of camouflaged data instances (10%, 25%, 50%, 75%, and 100%). This approach allows us to investigate how the distribution or frequency of camouflaged instances within the training data may affect the model's robustness and performance.

Including all three levels of camouflage in the training of these models aims to enhance their ability to generalize across a variety of adversarial conditions. However, given the potential impact of camouflage frequency on training, we maintain the percentage of camouflaged instances consistent for each individual model.

Furthermore, for each Camouflaged Model, we employ two different fine-tuning strategies, as depicted in Figure 1:

- **Static Modification**: This approach involves camouflaging the training dataset prior to model training. The benefit of this method lies in its simplicity and predictability, as the dataset remains consistent throughout the training process. However, this may limit the model's ability to adapt to new or varying types of camouflage not represented in the initial dataset.
- **Dynamic Modification**: In contrast, this method involves camouflaging the training dataset 'on-the-fly', introducing changes during the training process itself. This allows the model to be exposed to a wider variety and unpredictability of camouflage techniques, potentially improving its ability to generalize and adapt. The trade-off, however, is a more complex and computationally demanding training process.

Together, these different training and evaluation strategies provide a comprehensive approach to understanding the susceptibility of natural language processing models to word camouflage, and the potential strategies for improving their robustness. The parameters employed to the fine-tuning approaches can be found at 2.

3.2.2 External Validation

The AugLy library [19], developed by Meta AI, is used as an instrument of external validation in order to ensure the objectiveness and applicability of our evaluations. By generating an auxiliary test dataset, we can re-affirm our evaluations on model resilience against word camouflage. This library extends a distinctive approach of random text modifications, including letter substitutions with analogous Unicode or non-Unicode characters, punctuation insertions and font alterations. While these manipulations are based on random selection, instead of keyword extraction as in our methodology, they nonetheless simulate potential evasion techniques, providing a valuable counterpoint to our stratified camouflage techniques.

AugLy diverge from the 'pyleetspeak' package implemented in our research, which select word to be camouflaged based on semantic importance. Furthermore, AugLy exhibits a limited degree of flexibility in terms of user control and omits features such as word inversion and modification tracking.

Nevertheless, as an instrument of external validation, AugLy assumes a crucial role, especially when compared with the three-tier complexity levels. Comparative analysis with AugLy facilitates the identification of potential model

vulnerabilities and ensures unbiased results, ensuring a comprehensive defense strategy. This approach guarantees rigorous model evaluation and improves real-world performance resilience.

4 Experimental Setup

This section outlines the critical components of the research design. Subsection 4.1 details the specific architectures employed in the study, while the 'Data' subsection 4.2 describes the datasets used for adversarial sample creation and model training and evaluation. Lastly, subsection 4.3 establishes the evaluation metrics and procedures for determining model resilience against adversarial attacks.

4.1 Transformer Models

In this investigation, three distinct Transformer models are employed, each representing a unique configuration: encoder-only, decoder-only, and encoder-decoder. This selection facilitates the comparison of performance and resilience to adversarial attacks under diverse operational conditions.

BERT (bert-base-uncased) [32] serves as the encoder-only model. Renowned for its wide application and being among the most downloaded models on HuggingFace[2], BERT provides an ideal case for examining the potential susceptibility of prevalent models to adversarial onslaughts and word camouflage. Pretrained with the primary objectives of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), BERT boasts approximately 110M parameters.

The mBART model (mbart-large-50) [46] functions as the encoder-decoder paradigm. An extension of the original mBART model, it supports 50 languages for multilingual machine translation models. Its "Multilingual Denoising Pretraining" objective introduces noise to the input text, potentially augmenting its robustness against adversarial attacks. Developed by Facebook, this model encapsulates over 610M parameters.

Lastly, Pythia (pythia-410m-deduped) [47], forming part of EleutherAI's Pythia Scaling Suite, is harnessed. With its training on the Pile [48], a dataset recognized for its diverse range of English texts, it offers a fitting choice for analyzing model resilience to the often biased and offensive language pervasive on the internet. Comprising 410M parameters, it is well-equipped for this study.

4.2 Data

The study utilizes two primary datasets, OffensEval [49] and Constraint [50], representing distinct aspects of online behavior. OffensEval, part of the SemEval suite, consists of over 14,000 English tweets focusing on offensive language in social media. On the other hand, Constraint pertains to the detection of fake news related to COVID-19 across various social platforms, comprising a collection of 10,700 manually annotated posts and articles.

These datasets underscore the significance of combatting offensive language and misinformation, especially during critical times like a pandemic. They provide a solid foundation for examining the resilience of Transformer models to adversarial attacks and gauging the efficacy of camouflage techniques in evading detection.

To ensure data quality and reliability, several preprocessing steps were undertaken. These include eliminating duplicates, filtering out instances with fewer than three characters, preserving the original text case, and verifying the balance of the binary classes across both datasets. This balance verification is pivotal to ensure a fair assessment of camouflage techniques across different classes and avoid bias introduced by class imbalances.

Post-preprocessing, the datasets were divided into training, validation, and test sets, ensuring a comprehensive and balanced evaluation. For OffensEval and Constraint, the training set comprised 11,886 and 6,420 instances, respectively, with appropriate allocations for validation and testing.

4.3 Evaluating Robustness

For the evaluation of model performance, the F1-macro score is used. Despite the datasets being balanced, the choice of F1-macro score offers a more rigorous measure that enables the effective isolation of the impact of word camouflage techniques on performance and minimizes any potential bias due to class distribution.

A comprehensive experimental setup is implemented, encompassing 31 internal tests and an external test from AugLy. This detailed framework facilitates an in-depth assessment of how increasing complexity of camouflage techniques influence model performance and the practical application of the results.

[2] https://huggingface.co/models?pipeline_tag=fill-mask&sort=downloads

Table 3: Encoder-Only Transformer models original performance and performance reduction across camouflage levels, and AugLy. Models exceeding Naive ones are marked (*), with the least impacted by camouflage highlighted in bold.

Model	F1-Macro	Performance Reduction Levels							AugLy
		1.1	1.2	2.1	2.2	3.1	3.2	Avg	
Naïve	0.7782	2%	6%	6%	13%	7%	14%	8%	10%
10-static	0.7870*	2%	**5%**	6%	11%	6%	13%	7%	9%
25-static	0.7852*	2%	**5%**	6%	10%	**5%**	12%	**6%**	9%
50-static	0.7887*	2%	7%	**5%**	11%	**5%**	11%	7%	**8%**
75-static	0.7962*	3%	7%	**5%**	**9%**	**5%**	**10%**	7%	**8%**
100-static	0.4189	-	-	-	-	-	-	-	-
10-dynamic	0.7828*	3%	7%	5%	11%	**5%**	13%	7%	10%
25-dynamic	0.7947*	4%	8%	7%	12%	7%	13%	8%	10%
50-dynamic	0.7791	3%	8%	5%	9%	6%	**9%**	7%	8%
75-dynamic	0.7945*	3%	6%	5%	9%	6%	**9%**	6%	7%
100-dynamic	0.7527	**0%**	**4%**	**3%**	**8%**	**4%**	10%	**5%**	**5%**

(a) Encoder-only OffensEval comparative performance

Model	F1-Macro	Performance Reduction Levels							AugLy
		1.1	1.2	2.1	2.2	3.1	3.2	Avg	
Naïve	0.9677	4%	14%	7%	17%	11%	21%	12%	9%
10-static	0.9649	**1%**	3%	**1%**	6%	3%	6%	3%	7%
25-static	0.9649	**1%**	3%	2%	5%	2%	5%	3%	6%
50-static	0.9602	**1%**	3%	2%	4%	2%	4%	3%	**4%**
75-static	0.9517	**1%**	2%	**1%**	3%	2%	4%	**2%**	7%
100-static	0.9443	**1%**	**1%**	**1%**	3%	**1%**	**3%**	**2%**	8%
10-dynamic	0.9629	2%	3%	3%	5%	3%	6%	4%	6%
25-dynamic	0.9653	2%	3%	**1%**	5%	3%	5%	3%	6%
50-dynamic	0.9569	**1%**	2%	2%	4%	2%	5%	3%	7%
75-dynamic	0.9569	**1%**	2%	2%	3%	2%	4%	**2%**	6%
100-dynamic	0.9427	**1%**	**1%**	**1%**	**2%**	**1%**	**3%**	**2%**	**5%**

(b) Encoder-only Constraint comparative performance

Model robustness is assessed by the degree of performance reduction when models are evaluated on camouflaged test datasets at varied levels, relative to their performance on the original test dataset. The degree of performance reduction serves as a systematic measure of model resilience against adversarial attacks and illuminates how model performance deteriorates as difficulty levels increase and percentage of modified data escalates.

5 Experiment Results and Discussion

5.1 Phase I: Assessing Transformer Models' Susceptibility to Word Camouflage

Table 4: Decoder-Only Transformer models original performance and performance reduction across camouflage levels, and AugLy. Models exceeding Naive ones are marked (*), with the least impacted by camouflage highlighted in bold.

Model	F1-Macro	Performance Reduction Levels							AugLy
		1.1	1.2	2.1	2.2	3.1	3.2	Avg	
Naïve	0.7185	7%	15%	8%	16%	8%	16%	12%	11%
10-static	0.7159	8%	12%	9%	13%	8%	14%	11%	14%
25-static	0.6906	4%	11%	5%	12%	5%	11%	8%	8%
50-static	0.6750	4%	9%	5%	10%	6%	10%	7%	7%
75-static	0.6249	7%	9%	5%	7%	4%	8%	7%	8%
100-static	0.4309	**1%**	**1%**	**1%**	**1%**	**1%**	**1%**	**1%**	**0%**
10-dynamic	0.7351*	6%	13%	6%	15%	7%	15%	10%	9%
25-dynamic	0.7368*	5%	12%	7%	17%	9%	13%	11%	10%
50-dynamic	0.6636	3%	9%	4%	10%	5%	10%	7%	4%
75-dynamic	0.7029	7%	11%	8%	11%	7%	11%	9%	10%
100-dynamic	0.5932	**2%**	**4%**	**3%**	**8%**	**3%**	**6%**	**5%**	**5%**

(a) Decoder-only OffensEval comparative performance

Model	Test F1-Macro	Performance Reduction Levels							AugLy
		1.1	1.2	2.1	2.2	3.1	3.2	Avg	
Naïve	0.9380	4%	10%	8%	15%	7%	16%	10%	7%
10-static	0.9270	3%	11%	4%	12%	4%	11%	7%	8%
25-static	0.9046	3%	6%	4%	9%	4%	9%	6%	6%
50-static	0.8975	2%	5%	2%	6%	3%	7%	4%	4%
75-static	0.8700	3%	6%	3%	6%	3%	6%	5%	5%
100-static	0.7935	**1%**	**1%**	**1%**	**1%**	**1%**	**2%**	**1%**	**2%**
10-dynamic	0.9340	3%	7%	4%	8%	4%	9%	6%	8%
25-dynamic	0.9198	3%	9%	4%	12%	6%	11%	7%	6%
50-dynamic	0.8752	2%	5%	**2%**	6%	3%	6%	4%	**4%**
75-dynamic	0.8958	**1%**	**3%**	**2%**	**3%**	**2%**	**4%**	**2%**	**4%**
100-dynamic	0.3436	-	-	-	-	-	-	-	-

(b) Decoder-only Constraint comparative performance

The initial phase of the study is dedicated to investigating the vulnerability of Naive transformer models to adversarial attacks conducted through word camouflage. These models, having been trained on original datasets without prior exposure to word camouflage, are evaluated using the OffensEval and Constraint tasks. This assessment includes all three transformer configurations - Encoder-only, Decoder-only, and Encoder-Decoder, each being scrutinised independently.

A consistently emerging trend across all tasks and model configurations is the significant reduction in performance of Naive models as the complexity of camouflage levels intensifies.

As an illustration, in the OffensEval task, the Encoder-only Naive model's performance reduces by an average of 8%, starting from a 2% decrease at Level 1 and peaking at a 14% decrease at Level 3 (as depicted in Table 3a). The Decoder-only Naive models showcase a similar pattern, albeit with a steeper average performance decline of 12%,

Table 5: Comparative Performance and Resilience of Encoder-Decoder Transformer Models to Word Camouflage on the Offensive Language Task from (a) OffensEval and (b) Constraint. The 'Test F1-Macro' column indicates the F1-Macro score achieved by each model on the original, non-camouflaged Test dataset. Models outperforming the Naive model are marked with an asterisk (*). The 'Weakness' columns report the percentage reduction in performance on different levels of camouflaged Test datasets compared to the original Test dataset. In these columns, the model with the lowest percentage reduction is highlighted in bold, and the model with the greatest weakness is in italics. The 'AugLy' column indicates model performance on the AugLy camouflaged dataset.

Model	F1-Macro	Performance Reduction Levels							
		1.1	1.2	2.1	2.2	3.1	3.2	Avg	AugLy
Naïve	0.7436	6%	12%	7%	15%	7%	14%	10%	7%
10-static	0.7331	4%	9%	6%	10%	7%	13%	8%	10%
25-static	0.6330	4%	**5%**	**3%**	7%	**3%**	**7%**	**5%**	**5%**
50-static	0.7293	4%	10%	5%	9%	5%	10%	7%	15%
75-static	0.7282	**2%**	9%	**3%**	10%	4%	9%	6%	9%
100-static	0.6841	**2%**	8%	4%	**9%**	4%	9%	6%	7%
10-dynamic	0.7234	6%	11%	6%	12%	7%	13%	9%	11%
25-dynamic	0.7221	**2%**	9%	3%	15%	**2%**	11%	7%	**3%**
50-dynamic	0.7138	5%	9%	4%	10%	4%	11%	7%	7%
75-dynamic	0.6429	3%	4%	3%	7%	3%	7%	4%	9%
100-dynamic	0.6489	**2%**	**3%**	**1%**	**3%**	**2%**	**5%**	**3%**	5%

(a) Encoder-Decoder OffensEval comparative performance

Model	F1-Macro	Performance Reduction Levels							
		1.1	1.2	2.1	2.2	3.1	3.2	Avg	AugLy
Naïve	0.9568	5%	20%	7%	24%	10%	26%	15%	9%
10-static	0.9555	3%	7%	3%	8%	3%	10%	6%	7%
25-static	0.9415	2%	4%	2%	5%	2%	5%	3%	5%
50-static	0.9537	**1%**	2%	**1%**	**3%**	**1%**	**3%**	**2%**	7%
75-static	0.9489	**1%**	3%	**1%**	**3%**	**1%**	**3%**	**2%**	5%
100-static	0.9140	**1%**	**2%**	**1%**	**3%**	**1%**	**3%**	**2%**	**4%**
10-dynamic	0.9475	2%	5%	2%	6%	3%	7%	4%	5%
25-dynamic	0.9523	2%	4%	3%	6%	4%	6%	4%	6%
50-dynamic	0.9406	**1%**	4%	2%	5%	2%	5%	3%	**4%**
75-dynamic	0.9241	**1%**	3%	2%	4%	2%	4%	3%	**4%**
100-dynamic	0.9140	**1%**	**2%**	**1%**	**3%**	**1%**	**3%**	**2%**	**4%**

(b) Encoder-Decoder Constraint comparative performance

reaching a maximum of 16% at Level 3 (refer to Table 4a). Among all configurations, the Encoder-Decoder model exhibits the least drastic performance decline, with an average decrease of 10% and a maximum decrease of 10% at Level 3 (refer to Table 5a).

The same pattern is observed in the Constraint task, with the performance of the Naive models progressively deteriorates with the escalation of camouflage levels. The Encoder-only model, as presented in Table 3b, endures an average performance reduction of 12% across all levels, culminating in a 21% reduction at Level 3. In addition to this, the Decoder-only and Encoder-Decoder configurations experience average performance reductions of 10% and 15% respectively (Tables 4b, 5b).

These findings underscore the profound impact of increasing complexity on model performance, thereby spotlighting the inherent weaknesses that various evasion techniques can exploit. Intriguingly, when examining the performance reduction results for different configurations (refer to Tables 3, 4, and 5), it is evident that all models face heightened difficulties at Levels 2 and 3 compared to Level 1. Specifically, Level 3 and v2 (featuring an increased ratio of camouflaged words in each data instance) across all levels prove to be the most challenging.

Particularly, when more camouflaged words are present in a text (represented as v2), the model faces greater challenges than when fewer words are camouflaged (represented as v1). This conclusion is substantiated by the consistently elevated performance reduction percentages witnessed across all v2 levels compared to their v1 counterparts. It suggests that a seemingly straightforward strategy, such as the augmentation of words camouflaged within a text, can substantially compromise model performance.

Further evidence of this vulnerability is witnessed in the sharp decline in performance of the Naive model in the face of increasing percentages of camouflaged data. This is clearly depicted in the line plots corresponding to different Transformer configurations (Figures 3, 4 5, 6, 7 and 8). These figures, representing model performance against varying percentages of camouflaged data, mimic real-world scenarios. By simulating circumstances wherein users might employ a spectrum of complexity techniques in word camouflage, these plots offer insightful revelations into how the widespread implementation of such evasion techniques could influence model performance. They suggest potential shifts in performance as word camouflage techniques become more prevalent.

Figures 3 and 4 illustrate the Encoder-only Naive model's performance deterioration as the proportion of camouflaged data instances elevates within both OffensEval and Constraint tasks. A similar trend is discernible in the Decoder-only configuration (Figures 5 and 6) and in the Encoder-Decoder configuration (Figures 7 and 8).

In the following section, it will be demonstrated that this performance decline is more pronounced in Naive models compared to their adversarially-trained counterparts, further underscoring the susceptibility of Naive models to adversarial attacks.

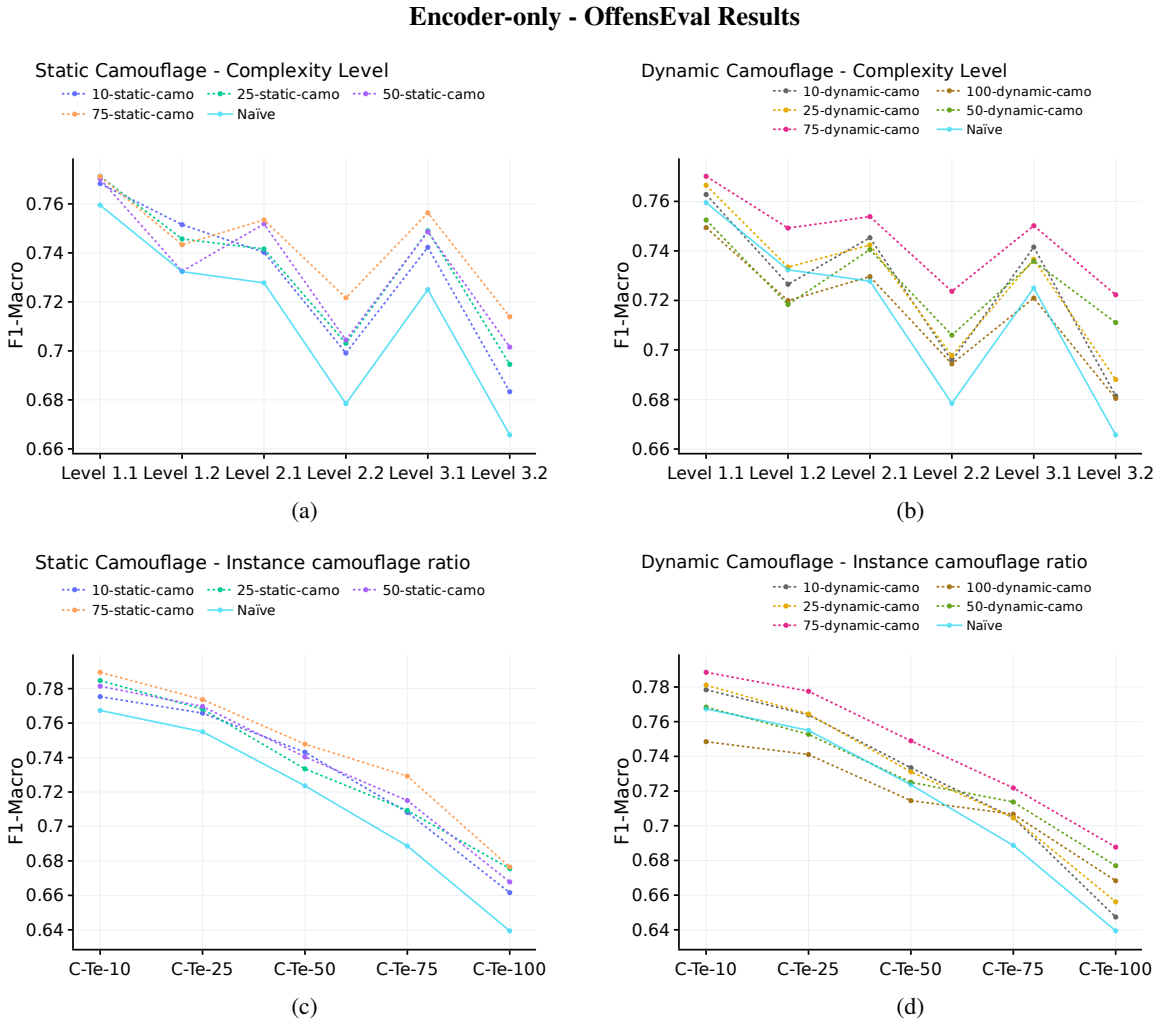


Figure 3: Comprehensive performance comparison of fine-tuned Encoder-only models against naive models in the Offensive Language task from OffensEval under various conditions. (a) Performance of Pre-camouflaged Models across different levels. (b) Performance of Var-camouflaged Models across different levels. (c) Performance of Pre-camouflaged Models across different camouflage percentages. (d) Performance of Var-camouflaged Models across different camouflage percentages.

Overall, these results bring to light the inherent susceptibility of Naive models across all transformer configurations to word camouflage attacks, thereby emphasizing the necessity for solutions aimed at fortifying these vulnerabilities.

5.2 Phase II: Enhancing Transformer Robustness Against Word Camouflage Attacks

The motivation behind the second phase of the study was to enhance the robustness of Transformer models against word camouflage attacks. In the first phase, it was determined that word camouflage attacks could significantly degrade the performance of the models. This phase aimed to evaluate the effectiveness of various countermeasures, mainly through the integration of adversarial training strategies using different proportions and methods of data camouflage.

5.2.1 Static vs Dynamic Camouflage

The analysis of the experiments revealed that models trained with a mix of original and statically camouflaged data, up to a 75% proportion, performed admirably across a variety language detection tasks in both the OffensEval and Constraint contexts. Specifically, in the OffensEval setting, the Encoder-only model trained with 75% statically camouflaged data demonstrated improving the performance and decreasing the performance reduction with complexity (see 3a). However, models trained on completely statically camouflaged datasets saw a significant drop in performance, with the

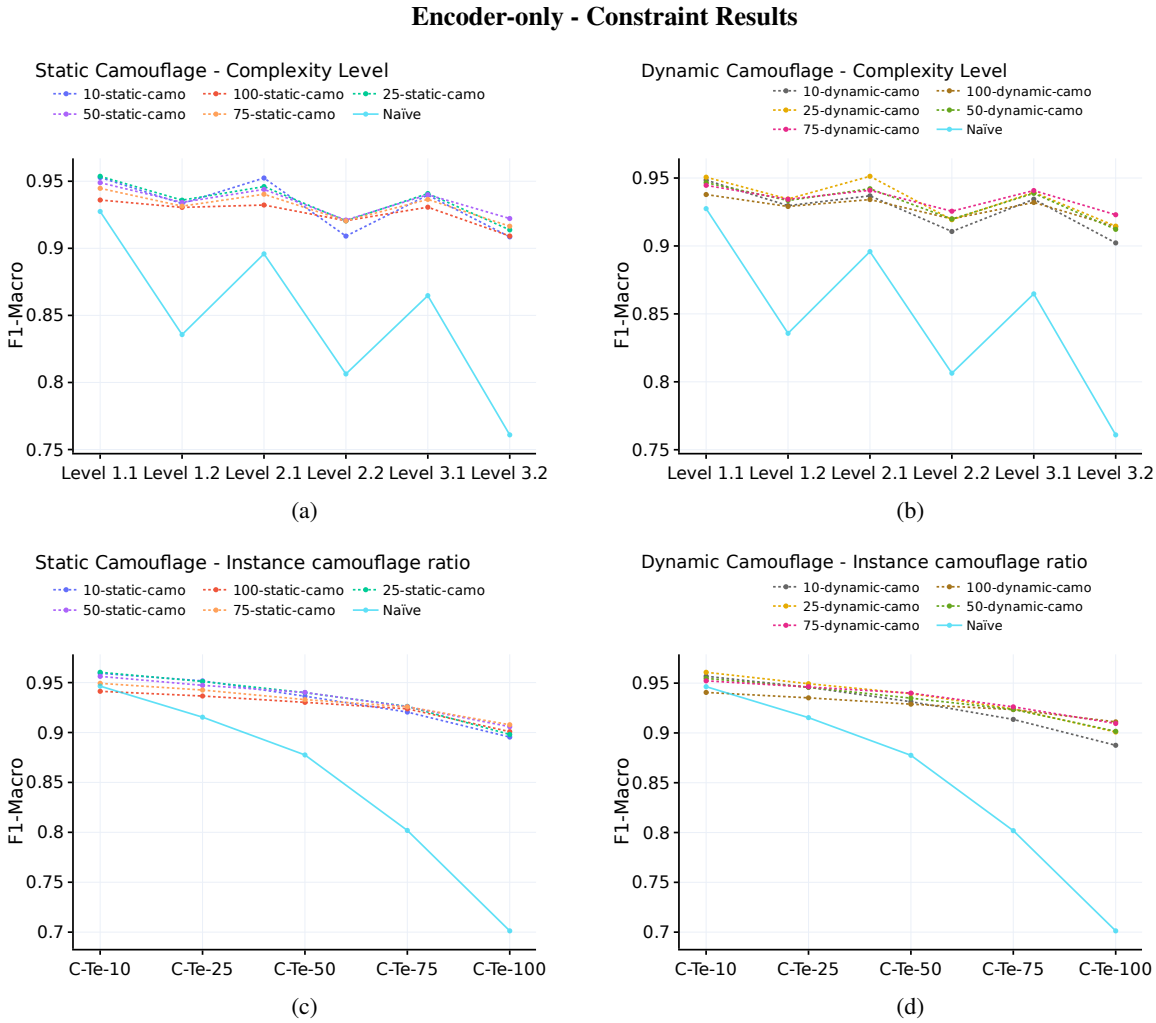


Figure 4: Comprehensive performance comparison of fine-tuned Encoder-only models against naive models in the False Information Language task from Constraint under various conditions. (a) Performance of Pre-camouflaged Models across different levels. (b) Performance of Var-camouflaged Models across different levels. (c) Performance of Pre-camouflaged Models across different camouflage percentages. (d) Performance of Var-camouflaged Models across different camouflage percentages.

F1-Macro score falling to 0.4189 across all levels for the '100-static' Encoder-only model (see 3a) and 0.3436 for the '100-dynamic' Decoder-only model (see 4b). This degradation in performance supports the conclusion that a balanced mix of original and camouflaged data produces superior results.

On the other hand, dynamic camouflage strategies displayed a distinct advantage, with models incorporating these techniques showing substantial robustness. For instance, a model that dynamically camouflaged 100% of the data during training managed to avoid performance deterioration and exhibited marked robustness in Table 3a while the 100% static counterpart stuck. This observation suggests that dynamic camouflage introduces a certain degree of variability and richness into the training data, which in turn enables the model to learn more effectively and flexibly, although it is still recommended to include some percentage of original, non-camouflaged data in the training process.

5.2.2 Effect of Camouflage Complexity

An intriguing trend was revealed when examining the relationship between the complexity of camouflage techniques and model performance. As the camouflage complexity level increased, naive models exhibited more pronounced performance reduction, as previously observed in Section 5.1 in both the OffensEval and Constraint tasks. This trend implies an inherent vulnerability of Naive models to heightened complexity in adversarial attacks. However,

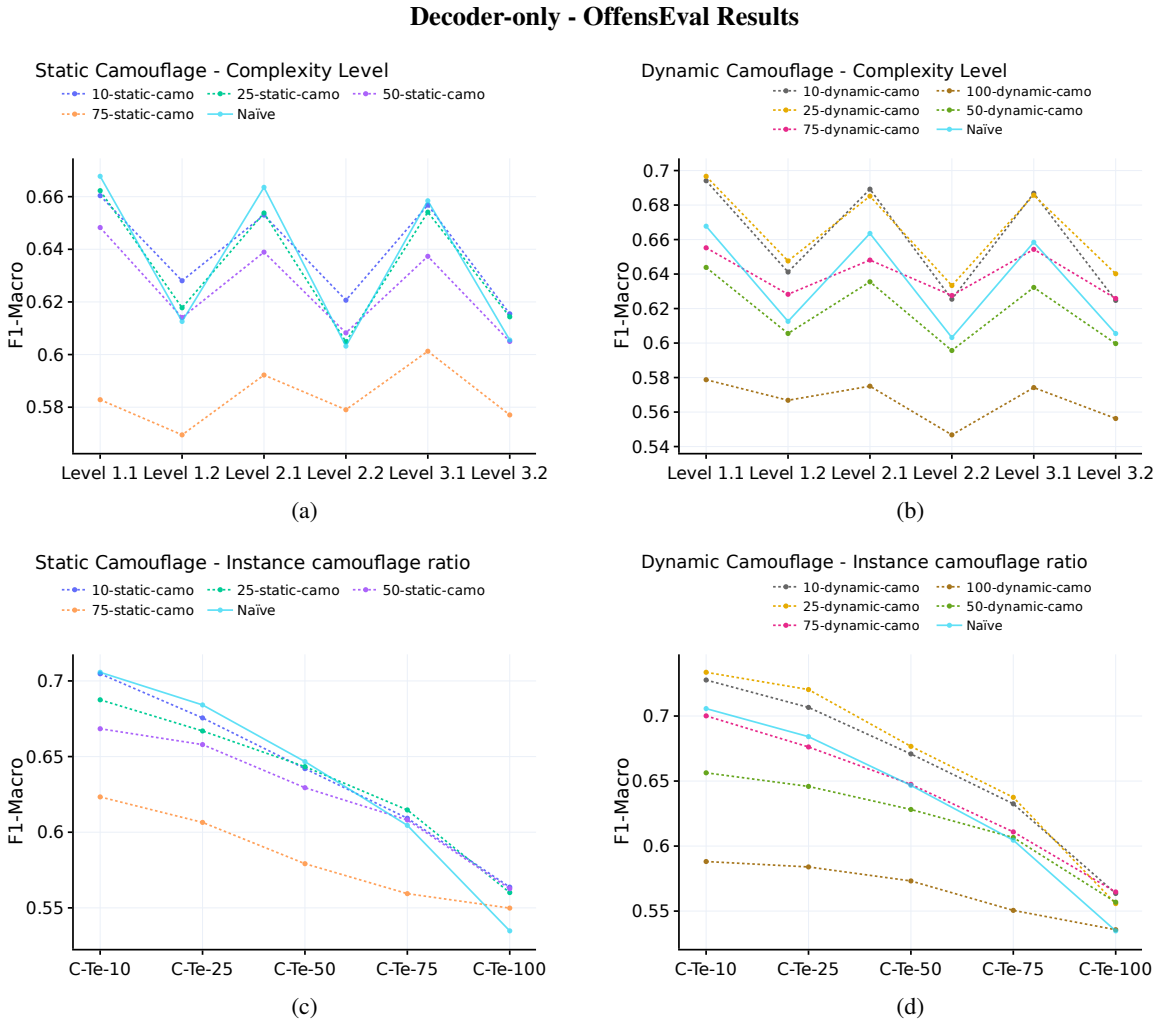


Figure 5: Comprehensive performance comparison of fine-tuned Decoder-only models against naive models in the Offensive Language task from OffensEval under various conditions. (a) Performance of Pre-camouflaged Models across different levels. (b) Performance of Var-camouflaged Models across different levels. (c) Performance of Pre-camouflaged Models across different camouflage percentages. (d) Performance of Var-camouflaged Models across different camouflage percentages.

adversarially trained models, particularly those trained with a combination of original and dynamic camouflage data, such as the '75-dynamic-camo' model, demonstrated lower degrees of reduction (Tables 3, 4 and 5), thereby highlighting the resilience of adversarially trained models and their enhanced ability to counter advanced camouflage attacks.

5.2.3 Influence of Camouflage Percentage

The analysis also brought to light an inverse relationship between the percentage of camouflaged data and model performance. As the proportion of camouflaged data in the test set increased, the model's performance correspondingly declined. The trend was evident across all models, indicating the importance of considering the expected degree of camouflage in the actual scenario in order to estimate the impact and to select the most appropriate adversarial training method.

5.2.4 Architectural Considerations

In addition to the above, an underlying trend emerged from the analysis, which transcended the specific architectural configurations: training with dynamic camouflage consistently demonstrated superior resilience to adversarial attacks.

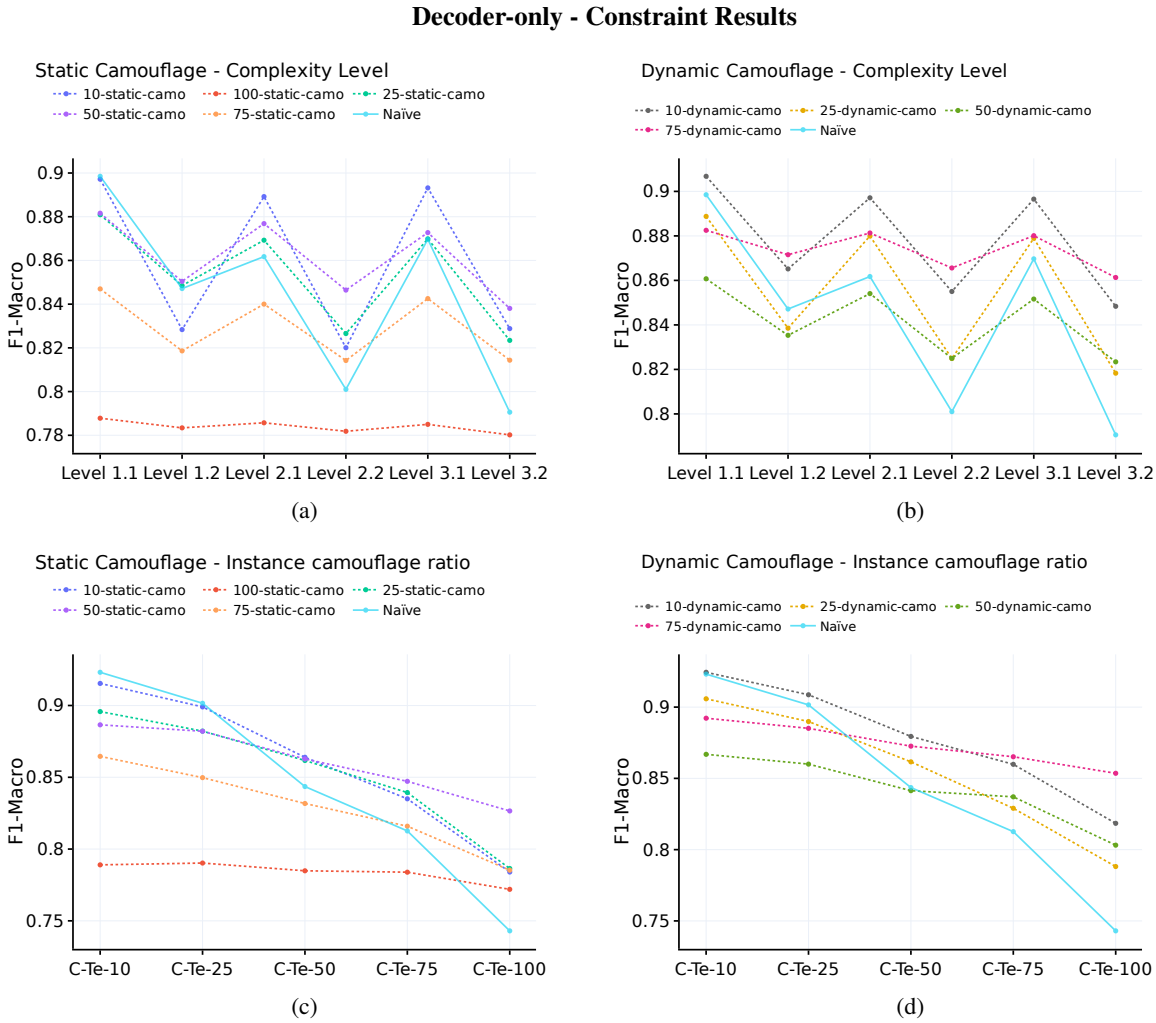


Figure 6: Comprehensive performance comparison of fine-tuned Decoder-only models against naive models in the False Information Language task from Constraint under various conditions. (a) Performance of Pre-camouflaged Models across different levels. (b) Performance of Var-camouflaged Models across different levels. (c) Performance of Pre-camouflaged Models across different camouflage percentages. (d) Performance of Var-camouflaged Models across different camouflage percentages.

This strategy maintained or even enhanced the original model's performance in many cases, while a high camouflage percentage brought significant challenges.

When assessing the impact of different model architectures, it was observed that Encoder-only models demonstrated significant robustness against adversarial attacks when trained with both static and dynamic camouflage techniques. In the Constraint task, for example, the '75-static-camo' and '25-static-camo' Encoder-only models outperformed their counterparts on several levels. On the other hand, Encoder-Decoder and Decoder-only setups also showcased improved performance, thus affirming that adversarial training strategies can be effectively utilized across diverse architectural setups.

Finally, a key finding was the inverse relationship between the proportion of camouflaged data in the test set and the model's performance. As the proportion of camouflaged data increased, there was a noticeable decline in performance across all models. Thus, striking a balance in data camouflaging is vital: while it can boost generalization, excessive usage may compromise performance.

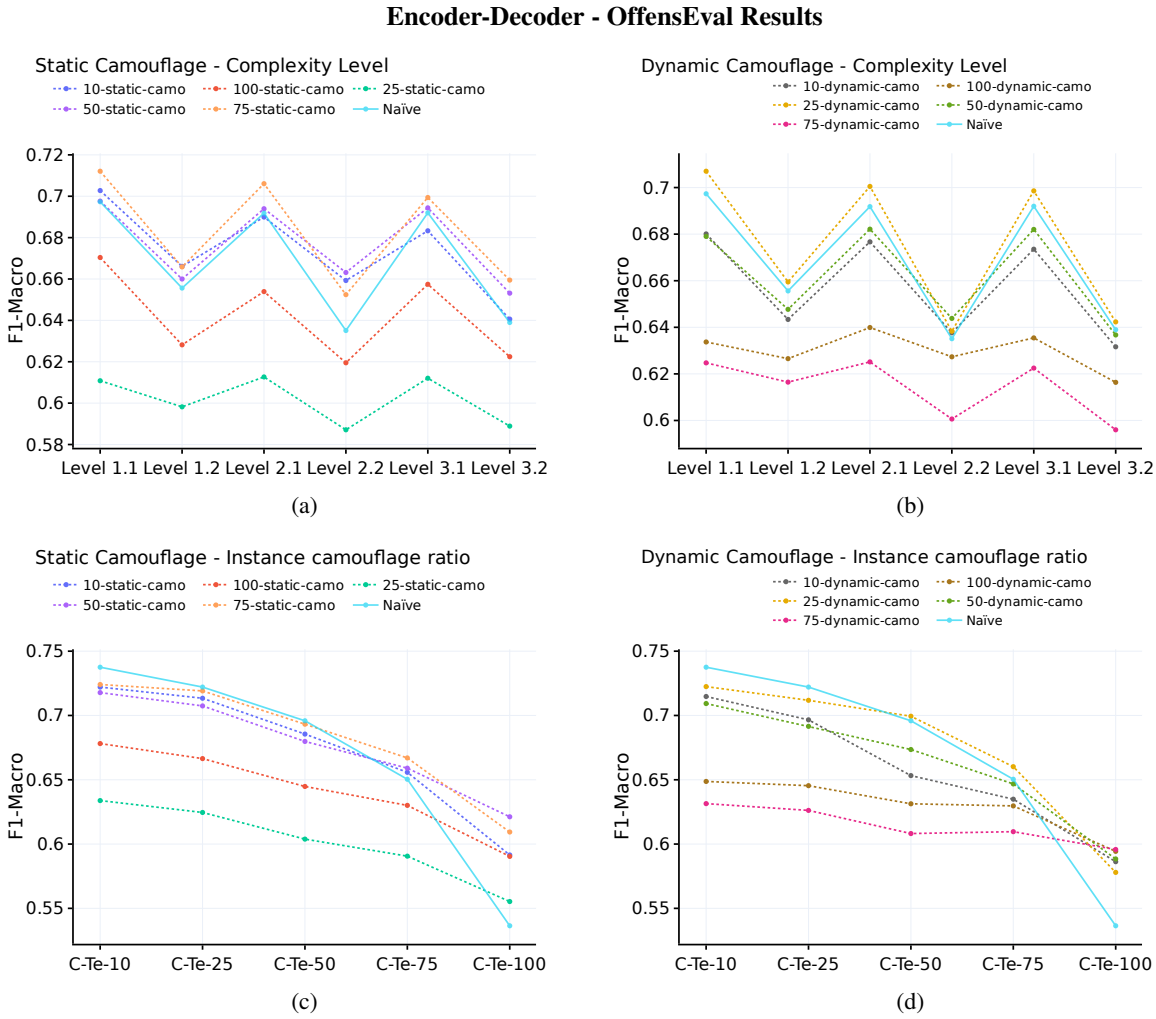


Figure 7: Comprehensive performance comparison of fine-tuned Encoder-Decoder models against naive models in the Offensive Language task from OffensEval under various conditions. (a) Performance of Pre-camouflaged Models across different levels. (b) Performance of Var-camouflaged Models across different levels. (c) Performance of Pre-camouflaged Models across different camouflage percentages. (d) Performance of Var-camouflaged Models across different camouflage percentages.

6 Conclusion

In response to the escalating prominence of adversarial attacks in Natural Language Processing (NLP), this research presented a two-phase methodology to enhance Transformer models' robustness. In the first phase, the susceptibility of naive models to camouflage adversarial attacks was identified, demonstrating a clear need for improved defences.

A proactive strategy, incorporating adversarial training with both static and dynamic camouflage, was introduced in the second phase. The models trained with a small proportion (10-25%) of statically camouflaged data outperformed those trained entirely with static camouflage. Dynamic camouflage introduced during training further boosted the models' learning and generalization capabilities.

In the comparison of various configurations, encoder-only models often excelled in managing adversarial attacks, underscoring their superior adaptability. However, all configurations faced difficulties as the proportion of camouflaged data increased, which emphasized the importance of balancing between original and camouflaged data in the training set.

These findings carry significant implications for improving AI system robustness. Nevertheless, the limitations of this study are acknowledged. The proposed approach's effectiveness may fluctuate based on camouflage complexity and

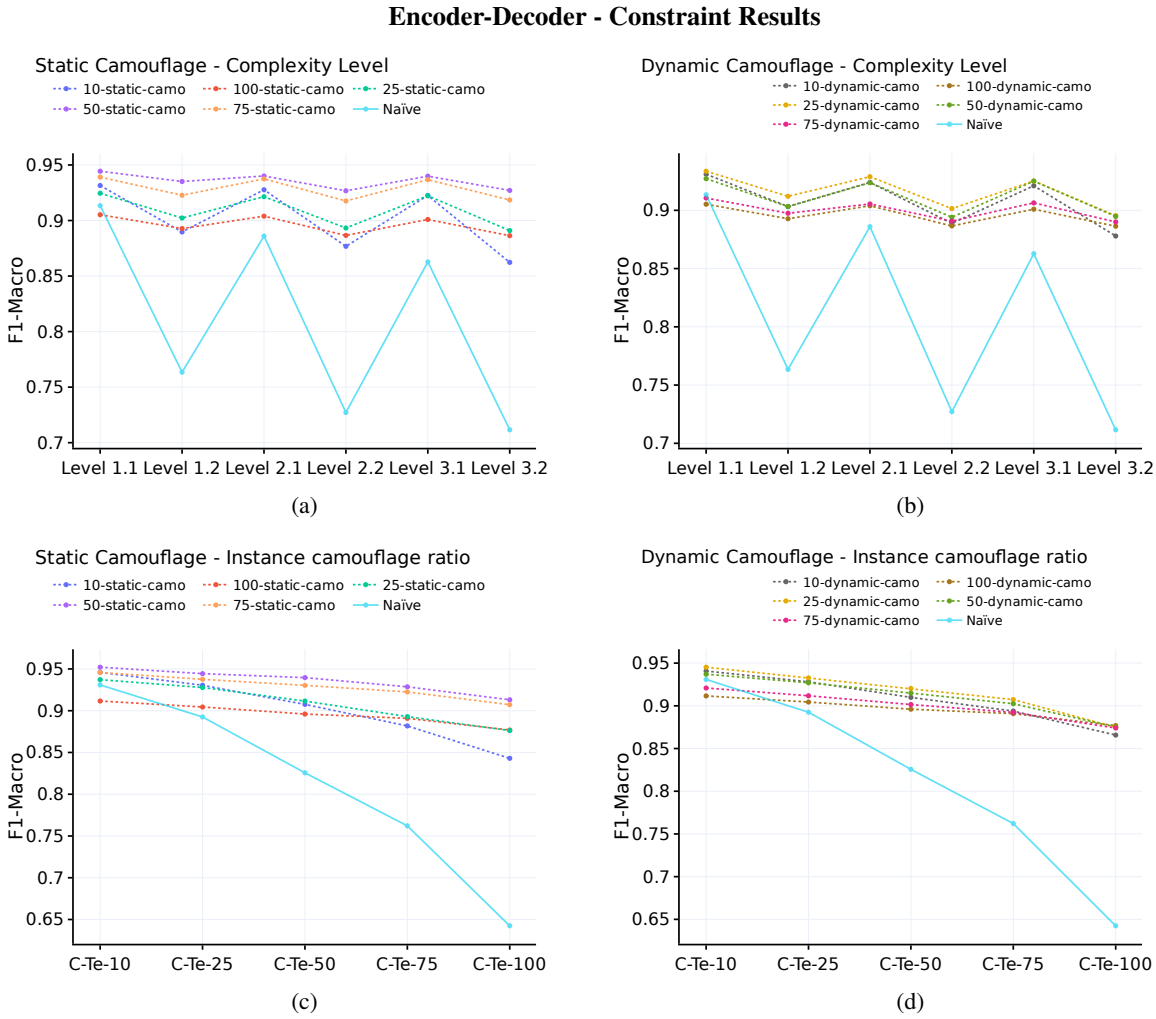


Figure 8: Comprehensive performance comparison of fine-tuned Encoder-Decoder models against naive models in the False Information Language task from Constraint under various conditions. (a) Performance of Pre-camouflaged Models across different levels. (b) Performance of Var-camouflaged Models across different levels. (c) Performance of Pre-camouflaged Models across different camouflage percentages. (d) Performance of Var-camouflaged Models across different camouflage percentages.

the type of data encountered. Moreover, this research primarily concentrated on black-box adversarial attacks, leaving other types largely unexplored.

Future research could expand this methodology to other adversarial attack types and model architectures, and further explore the influence of camouflage complexity and type on model learning and robustness.

References

[1] T. G. Dietterich, Steps toward robust artificial intelligence, AI Magazine 38 (3) (2017) 3–24. doi:10.1609/aimag.v38i3.2756.

[2] S. Patil, V. Varadarajan, D. Walimbe, S. Gulechha, S. Shenoy, A. Raina, K. Kotecha, Improving the Robustness of AI-Based Malware Detection Using Adversarial Machine Learning, Algorithms 14 (10) (2021) 297. doi:10.3390/a14100297.

[3] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, C. Li, Adversarial attacks on deep-learning models in natural language processing: A survey, ACM Trans. Intell. Syst. Technol. 11 (3) (2020). doi:10.1145/3374217.

[4] E. Kotei, R. Thirunavukarasu, A systematic review of transformer-based pre-trained language models through self-supervised learning, Information 14 (3) (2023). doi:10.3390/info14030187.

[5] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.

[6] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami, Practical black-box attacks against machine learning, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 506–519. doi:10.1145/3052973.3053009.

[7] R. Jia, P. Liang, Adversarial examples for evaluating reading comprehension systems, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2021–2031. doi:10.18653/v1/D17-1215.

[8] W. H. Organization, et al., Infodemic management: an overview of infodemic management during covid-19, january 2020–may 2021 (2021).

[9] A. Martín, J. Huertas-Tato, Á. Huertas-García, G. Villar-Rodríguez, D. Camacho, FacTeR-Check: Semi-automated fact-checking through semantic similarity and natural language inference, Knowledge-Based Systems 251 (2022) 109265. doi:10.1016/j.knosys.2022.109265.

[10] G. Ruffo, A. Semeraro, A. Giachanou, P. Rosso, Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language, Computer Science Review 47 (2023) 100531. doi:https://doi.org/10.1016/j.cosrev.2022.100531.

[11] W. Xu, Y. Qi, D. Evans, Automatically evading classifiers: A case study on pdf malware classifiers, in: Network and Distributed System Security Symposium, 2016.

[12] J. Jeong, S. Kwon, M.-P. Hong, J. Kwak, T. Shon, Adversarial attack-based security vulnerability verification using deep learning library for multimedia video surveillance, Multimedia Tools and Applications 79 (23-24) (2020) 16077–16091. doi:10.1007/s11042-019-7262-8.

[13] C.-L. Chang, J.-L. Hung, C.-W. Tien, C.-W. Tien, S.-Y. Kuo, Evaluating robustness of ai models against adversarial attacks, in: Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence, SPAI '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 47–54.

[14] M. Kavanagh, Bridge the generation gap by decoding leetspeak, Inside the Internet 12 (12) (2005) 11.

[15] A. Romero-Vicente, Word camouflage to evade content moderation (2021).
URL https://www.disinfo.eu/publications/word-camouflage-to-evade-content-moderation/

[16] K. Blashki, S. Nichol, Game geek's goss: linguistic creativity in young males within an online university forum, 2005. doi:10536/DRO/DU:30003258.

[17] J. Fuchs, Gamespeak for n00bs - a linguistic and pragmatic analysis of gamers' language, Ph.D. thesis, University of Graz (2013).
URL https://unipub.uni-graz.at/obvugrhs/content/titleinfo/231890?lang=en

[18] R. Craenen, Leet speak cheat sheet.
URL https://www.gamehouse.com/blog/leet-speak-cheat-sheet/

[19] Z. Papakipos, J. Bitton, Augly: Data augmentations for robustness (2022). arXiv:2201.06494.

[20] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples (2015). arXiv:1412.6572.

[21] M. Cheng, J. Yi, P.-Y. Chen, H. Zhang, C.-J. Hsieh, Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples (2020). arXiv:1803.01128.

[22] P. Michel, X. Li, G. Neubig, J. Pino, On evaluation of adversarial perturbations for sequence-to-sequence models, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3103–3114. doi:10.18653/v1/N19-1314.

[23] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, K.-W. Chang, Generating natural language adversarial examples, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2890–2896. doi:10.18653/v1/D18-1316.

[24] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey, Ieee Access 6 (2018) 14410–14430.

[25] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 2154–2156.

[26] S. Singh, A. Mahmood, The nlp cookbook: Modern recipes for transformer based deep learning architectures, IEEE Access 9 (2021) 68675–68702. doi:10.1109/ACCESS.2021.3077350.

[27] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (6088) (1986) 533–536.

[28] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 28, Curran Associates, Inc., 2015.

[29] D. W. Otter, J. R. Medina, J. K. Kalita, A survey of the usages of deep learning for natural language processing, IEEE transactions on neural networks and learning systems 32 (2) (2020) 604–624.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need (2017). arXiv:1706.03762.

[31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer (2020). arXiv:1910.10683.

[32] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019.

[34] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners (2020). arXiv:2005.14165.

[35] B. Liang, H. Li, M. Su, P. Bian, X. Li, W. Shi, Deep text classification can be fooled, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2018. doi:10.24963/ijcai.2018/585.

[36] M. Blohm, G. Jagfeld, E. Sood, X. Yu, N. T. Vu, Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 108–118. doi:10.18653/v1/K18-1011.

[37] C. Guo, A. Sablayrolles, H. Jégou, D. Kiela, Gradient-based adversarial attacks against text transformers, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 5747–5757. doi:10.18653/v1/2021.emnlp-main.464.

[38] R. Maheshwary, S. Maheshwary, V. Pudi, A context aware approach for generating natural language attacks (2020). arXiv:2012.13339.

[39] Y. Gil, Y. Chai, O. Gorodissky, J. Berant, White-to-black: Efficient distillation of black-box adversarial attacks (2019). arXiv:1904.02405.

[40] M. Li, J. Yu, S. Li, J. Ma, H. Liu, Textual adversarial attacks on named entity recognition in a hard label black box setting, 2022 15th International Conference on Advanced Computer Theory and Engineering (ICACTE) (2022) 55–60.

[41] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, M. Kim, An analysis of adversarial attacks and defenses on autonomous driving models (2020). arXiv:2002.02175.

[42] K. Chiyomaru, K. Takemoto, Adversarial attacks on voter model dynamics in complex networks, Phys. Rev. E 106 (2022) 014301. doi:10.1103/PhysRevE.106.014301.
URL https://link.aps.org/doi/10.1103/PhysRevE.106.014301

[43] M. F. Chen, M. Z. Rácz, An adversarial model of network disruption: Maximizing disagreement and polarization in social networks, IEEE Transactions on Network Science and Engineering 9 (2022) 728–739.

[44] X. Yin, W. Lin, K. Sun, C. Wei, Y. Chen, A2s2-gnn: Rigging gnn-based social status by adversarial attacks in signed social networks, IEEE Transactions on Information Forensics and Security 18 (2023) 206–220.

[45] Álvaro Huertas-García, A. Martín, J. H. Tato, D. Camacho, Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflage (2022). arXiv:2212.14727.

[46] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, A. Fan, Multilingual translation with extensible multilingual pretraining and finetuning (2020). arXiv:2008.00401.

[47] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, O. van der Wal, Pythia: A suite for analyzing large language models across training and scaling (2023). arXiv:2304.01373.

[48] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy, The pile: An 800gb dataset of diverse text for language modeling (2020). arXiv:2101.00027.

[49] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval), in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 75–86. doi:10.18653/v1/S19-2010.

[50] P. Patwa, S. Sharma, S. Pykl, V. Guptha, G. Kumari, M. S. Akhtar, A. Ekbal, A. Das, T. Chakraborty, Fighting an infodemic: COVID-19 fake news dataset, in: Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer International Publishing, 2021, pp. 21–29. doi:10.1007/978-3-030-73696-5_3.

Chapter 5

General Discussion

5.1 Discussions and Results and Contributions

This thesis presents a comprehensive exploration of challenges and solutions in multilingual NLP, particularly focusing on enhancing the robustness and efficiency of transformer models in the context of misinformation detection and content moderation. The research consists of three interconnected studies to address critical aspects of NLP in an increasingly complex digital landscape.

The first study lays the foundation by investigating dimensionality reduction techniques for multilingual transformer models, aiming to optimize their performance and efficiency in semantic textual similarity tasks. This work directly feeds into the second study, which leverages the use of semantic-awareness fine-tuned models to develop advanced techniques for detecting word camouflage in online social networks, a common tactic used to evade content moderation. Building on these insights, the final study concludes this research by examining how different Transformer architecture models and their tokenization preprocessing are susceptible to content evasion techniques and proposing novel adversarial training strategies to enhance their intrinsic robustness.

Together, these studies form a cohesive narrative that addresses the multiple challenges of developing resilient, efficient, and effective NLP systems for countering misinformation and improving content moderation in a multilingual context.

For the sake of clarity, we will wrap the main contributions and discussion from these main articles associated with the thesis research.

5.2 Enhancing Multilingual Transformer Models

5.2.1 Dimensionality Reduction in NLP

Dimensionality reduction techniques are crucial in Natural Language Processing (NLP) for managing the high-dimensional embeddings produced by transformer models. These

techniques aim to reduce computational overhead and storage requirements while preserving critical information necessary for effective language understanding and generation. This first study evaluates a range of unsupervised dimensionality reduction techniques on multilingual transformer models, focusing on their impact on Semantic Textual Similarity (STS) tasks. Details about the methodology of this research can be consulted in Sections 3.1.4, 3.6, and 3.7.1; and published article 4.1.

Transformer models, especially those designed for multilingual understanding, generate high-dimensional embeddings that represent textual information. These embeddings capture a wide array of semantic nuances but at a cost of increased computational complexity. Dimensionality reduction compresses these embeddings into a more manageable size, reducing the computational resources required for storage and processing while aiming to preserve as much of the original semantic information as possible.

Processing high-dimensional embeddings demands significant computational power and memory, which can be limiting, especially in resource-constrained environments. By reducing the dimensionality of these embeddings, the study aims to make the semantic analysis process more efficient, including speeding up operations such as similarity computation between embeddings, which is central to STS tasks.

5.2.2 Evaluation of Techniques on Pre-trained Models

The application of dimensionality reduction techniques to pre-trained embeddings revealed several important findings. Notably, ICA emerged as the most effective technique, achieving an average dimensionality reduction of 91.58% while improving performance in the STS task. This significant reduction highlights the potential of ICA to optimize the feature space of pre-trained models, making them more efficient without compromising their performance. ICA's ability to handle noisy and unadjusted variables in pre-trained embeddings contributed to its superior performance compared to other techniques like PCA and KPCA.

5.2.3 Evaluation of Techniques on Fine-tuned Models

For fine-tuned embeddings, the study demonstrated that dimensionality reduction techniques could still achieve considerable reductions while maintaining or even enhancing performance. KPCA with the sigmoid kernel was particularly effective, indicating that managing nonlinearity becomes more critical after fine-tuning. The variance threshold technique also showed promising results, benefiting from the reduced presence of noisy variables due to the fine-tuning process. On average, a 54.65% reduction in dimensions was achieved, illustrating that even fine-tuned models can be significantly optimized for efficiency.

5.2.4 Comparative Analysis and Performance Improvement

The comparative analysis of different techniques underscored the trade-offs between dimensionality reduction and computational efficiency. ICA stood out for pre-trained models, offering the best balance between dimensionality reduction and performance improvement. In contrast, KPCA and variance threshold techniques were more effective for fine-tuned models. The

Model	Ap. 1 r_s	Best Technique	Dimensions	Ap. 3 r_s	Fitting Time
bert-base-multilingual-cased	0.4342	ICA	209	0.5019	4m 16s
distilbert-base-multilingual-cased	0.4531	ICA	169	0.523	2m 47s
xlm-roberta-base	0.3274	ICA	249	0.5269	7m 51s
xlm-roberta-large	0.2855	ICA	1024	0.5392	31m 22s
LaBSE	0.7096	ICA	129	0.7488	2m 27s

Table 5.1: Average Spearman r_s correlation coefficient comparison between Approach 1 (Ap. 1) and best dimensional reduction technique in Approach 3 (Ap. 3) for the multilingual Transformers.

efficiency gains were substantial, with reduced embeddings leading to faster computations and lower storage requirements. For instance, dimensionality reduction techniques resulted in a fitting time that was 96.68% faster than the fine-tuning process, highlighting the practical benefits of these methods.

5.2.5 Dimensionality Reduction Impact

Approach 1 vs. Approach 3

Dimensionality reduction significantly improved the performance of pre-trained embeddings. ICA, in particular, was effective, retaining and even enhancing the performance of pre-trained embeddings by reducing dimensions by an average of 91.58% $\pm$ 2.59% while maintaining 100% of the baseline performance.

The reason for tat relies in the core objective of dimensionality reduction in this context is not just to shrink the size of the data but to distill the embeddings to their most informative features for semantic analysis. Additionally, high-dimensional spaces can lead to overfitting, where a model learns noise in the training data as if it were a significant pattern, compromising its ability to generalize to new, unseen data. Dimensionality reduction can help mitigate this risk by simplifying the models' representations and helping them generalize better from the training data to real-world semantic tasks.

Approach 2 vs. Approach 4

Fine-tuning combined with dimensionality reduction (Approach 4) showed a consistent improvement in performance over Approach 2. However, the degree of improvement varied by model and technique. ICA remained effective but less dominant compared to its performance with pre-trained embeddings. Nonlinear techniques such as KPCA-sigmoid also showed strong performance, particularly for complex models like xlm-roberta-large. Execution Time:

Dimensionality reduction techniques offered substantial time savings compared to fine-tuning. ICA, while being the most time-consuming among the reduction techniques, still offered

Model	Ap. 2 r_s	Best Technique	Dimensions	Ap. 4 r_s	Fitting Time
bert-base-multilingual-cased-fine-tuned	0.7045	ICA	568	0.7117	12m 38s
distilbert-base-multilingual-cased-fine-tuned	0.6863	VarThres	692	0.6842	2s
xlm-roberta-base-fine-tuned	0.7470	VarThres	673	0.7495	3s
xlm-roberta-large-fine-tuned	0.8150	KPCA-sigmoid	1024	0.8176	20m 6s
LaBSE-fine-tuned	0.8242	KPCA-sigmoid	768	0.8243	19m 25s

Table 5.2: Average Spearman r_s correlation coefficient comparison between Approach 2 (Ap. 2) and best dimensional reduction technique in Approach 4 (Ap. 4) for the multilingual Transformers.

a significant reduction in computational time compared to the fine-tuning process, by an average of 96.68% ± 0.68%.

5.2.6 Implications for Multilingual NLP

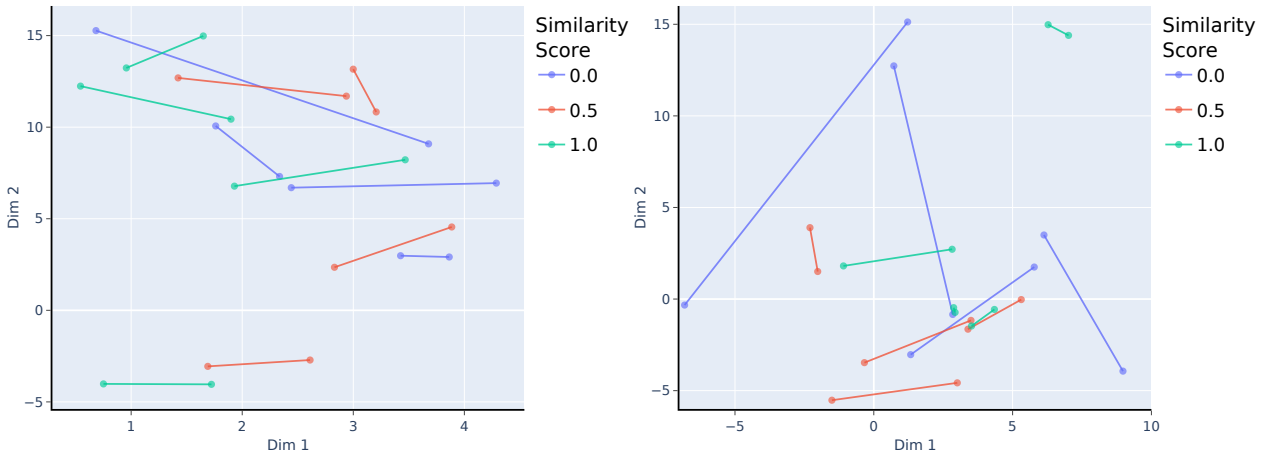
The study's focus on multilingual transformer models addressed the critical issue of the language bottleneck in NLP. By demonstrating the efficacy of dimensionality reduction techniques in a multilingual context, the research highlighted the potential to enhance NLP models' applicability and performance across diverse languages. This is particularly important for extending NLP capabilities to low-resource languages, ensuring that advanced language technologies are accessible and effective globally.

In multilingual contexts, capturing and comparing semantics across languages introduces additional complexity, given the vast and nuanced differences between languages. Dimensionality reduction techniques can help create more compact, language-agnostic representations of text that still retain the essential semantic features necessary for accurate cross-lingual comparison and analysis.

Another remarkable outcome is that feature extraction techniques proved to be more useful than feature selection since transforming the initial high-dimensional space into a new reduced latent space can extract more information rather than selecting specific variables, even if these variables are properly adjusted to the task.

5.2.7 Embeddings Visualization and Interpretability

Reducing the dimensionality of embeddings also aids in exploratory data analysis and visualization of high-dimensional data checking the effect of semantic training on the latent



(a) (b)

Figure 5.1: Comparison of the representation of embeddings generated by the IPCA technique for the bert-base-multilingual-cased model in Approach 3 (a) and Approach (4) of a representative set of sentence pairs from the STS test split with different levels of semantic similarity.

space of the models. Techniques like UMAP preserve local and global structures within the data and enable visual representation of semantic relationships, providing insights into how models perceive semantic similarities across languages. For instance, Mikolov et al. (2013) explored the projections of high-dimensional word embeddings in 2D to extract multiple relationships between words. For this reason, in this work, we have also applied the different techniques in 2D and 3D dimension reduction to study their usefulness for visualization and model interpretability.

A straightforward example of the applicability of these techniques can be found in Figure 5.1. In this Figure, we compare the 2D embeddings representation of a random subset of pairs of sentences of the STS dataset in the English language with different levels of semantic similarity. The comparison between pre-trained and fine-tuned embeddings allows us to observe the effect of fine-tuning the bert-base-multilingual-cased model. In the pre-trained case, we observe how the distance between different pairs of sentences is equal and does not match the labeled similarity. However, after fine-tuning, we observe a better agreement between the spatial latent representation and the labeled similarity, demonstrating and supporting the improvement of the latent space for the downstream task fine-tuning process.

5.2.8 Trade-offs and Recommendations

The choice of dimensionality reduction technique depends on the balance between desired performance, computational resources, and execution time. ICA is recommended for scenarios prioritizing dimensionality reduction with high performance retention, especially for pre-trained embeddings. For fine-tuned embeddings, nonlinear techniques like KPCA and feature selection methods like variance threshold offer valuable alternatives.

5.2.9 Impact & Future Directions

This study provided valuable insights into the impact of various dimensionality reduction techniques on multilingual transformer models. The findings emphasized the potential of techniques like ICA and KPCA to optimize the feature space, enhance performance, and improve computational efficiency. Future research could explore additional dimensionality reduction methods, apply these techniques to other NLP tasks, and further investigate their applicability in multilingual contexts. The ongoing development in the field is progressively steering towards the creation of universal models capable of handling multiple languages and modalities with minimal resources, showcasing the potential for zero-shot transfers and significant performance improvements even without labeled data. This research aligns with the motivation and research questions of the thesis, underscoring the importance of selecting appropriate dimensionality reduction techniques to balance performance and computational efficiency, ultimately contributing to more robust and scalable NLP solutions.

This research aligns with and addresses several ongoing challenges in the field of multilingual NLP. For instance, the "Curse of Multilinguality" and the "Embedding Barrier" [120] emphasize the difficulties in managing high-dimensional embeddings in multilingual models. Our approach of using dimensionality reduction techniques addresses these challenges by effectively managing and optimizing high-dimensional embeddings, thus contributing to more efficient and scalable multilingual models.

The "Curse of Multilinguality" refers to the inability of multilingual models to represent all languages equally due to the underrepresentation of low-resource languages in the pretraining corpus and the limited capacity of the model. As more languages are added, the model's performance can degrade after a certain point, highlighting the difficulty in creating truly universal multilingual models.

The "Embedding Barrier", on the other hand, stems from the high imbalance in the pretraining corpus, leading to very limited representation of low-resource language vocabularies in multilingual models. This issue is particularly severe for languages that do not share vocabulary or script with high-resource languages, resulting in poor tokenization and increased input sequence lengths. Consequently, words in low-resource languages are often tokenized into many subwords, hindering model learning and making training computationally expensive.

The work on dimensionality reduction techniques addresses these challenges by optimizing the high-dimensional embeddings produced by multilingual transformer models. By creating more compact, language-agnostic representations of text that retain essential semantic features, our approach mitigates the computational overhead associated with processing high-dimensional embeddings, particularly for low-resource languages with limited vocabulary representation.

Furthermore, our findings demonstrate that dimensionality reduction techniques like ICA and KPCA can optimize the feature space of multilingual transformer models while enhancing or maintaining their performance on tasks like Semantic Textual Similarity (STS). This not only improves computational efficiency but also contributes to more robust and scalable multilingual NLP solutions, potentially alleviating the "Curse of Multilinguality" by enabling better representation and performance across diverse languages, including low-resource ones.

Finally, the findings and techniques developed in this study have been further exploited in other research projects that this tehsis has contributed on: FacTeR-Check [243]. FacTeR-Check is a multilingual architecture for semi-automated fact-checking and hoax propagation analysis designed for both the general public and fact-checking organizations. This system applies insights from our research on dimensionality reduction in the feature space of multilingual transformer models to develop semi-automated fact-checking tools, natural language inference, and content search query building through automatic keyword extraction to extract and determine with entailment the veracity of a claim. It demonstrates the practical use of feature space dimensionality reduction techniques in semantic search for effective misinformation detection across languages, showcasing the real-world applicability and impact of our work.

In conclusion, this research has demonstrated the significant potential in addressing the challenges posed by high-dimensional embeddings in multilingual transformer models. Through the effective application of dimensionality reduction techniques, we have demonstrated the ability to handle high-dimensional data while improving model efficiency and enhancing semantic analysis capabilities. By distilling embeddings to their most informative features, our approach mitigates the risk of overfitting and enables better generalization, contributing to more robust multilingual representations. Furthermore, the dimensionality reduction techniques facilitate exploratory data analysis and visualization, providing valuable insights into how models perceive semantic similarities across languages. Furthermore, our techniques facilitate more inclusive multilingual representation by creating compact, language-agnostic representations that retain essential semantic information, addressing the "Embedding Barrier" and contributing to the development of truly universal models. The real-world applicability of our work is exemplified by its integration into projects like FacTeR-Check, a multilingual architecture for semi-automated fact-checking and hoax propagation analysis, demonstrating a more realistic way of countering false information by leveraging optimized multilingual models.

5.3 Detection of Word Camouflage in Online Social Networks

As presented in Section 3.1.5, content moderation on online social networks is critical to maintaining a safe environment and preventing the spread of harmful content, including misinformation, hate speech, and other malicious behaviors. However, as content moderation techniques evolve, so do the evasion strategies employed by malicious actors. The second article associated within the thesis outlines the critical findings from the research paper published in Applied Soft Computing [181], highlighting the development and utilization of a synthetic multilingual dataset and the Python package "pyleetspeak" to address the issue of content evasion in social networks.

Two main contributions can be highlighted from this research. Firstly, the development of a customizable, multilingual methodology for simulating content evasion used for the generation of a synthetic multilingual dataset of camouflaged words, annotated with examples of word camouflage in English, Spanish, French, Italian, and German. Secondly, the development of Named Entity Recognition multilingual models for word camouflage detection, where we

```
OrderedDict([('sentence',
              'This is an example of leetspeak text for NER data generation'),
             ('meta',
              [{'kw': 'leetspeak',
                'init_idxs': (22, 31),
                'kw_leet': 'l£;@tspeak',
                'params': {'leetspeak-covid_basic': {'change_prb': 0.8,
                  'change_frq': 0.5,
                  'mode': 'covid_basic',
                  'get_all_combs': False,
                  'uniform_change': array(False),
                  'seed': 20,
                  'list_changes': [('a', ['@', '4', 'Δ', '*', '', '.', ' ']),
                   ('e', ['3', '€', '£', '%', '@', '*', '', '.', ' ']),
                   ('i', ['1', 'l', '¡', '!', "'", '*', '', '.', ' ']),
                   ('o', ['0', 'ø', '*', '', '.', ' ']),
                   ('oo', ['u', ' ']),
                   ('u', ['_', 'Ü', 'ü', '*', '.', ' '])],
                  'text_in': 'leetspeak',
                  'text_out': 'l£@tspeak'},
                 'punct_camo': {'seed': 20,
                  'uniform_change': array(True),
                  'hyphenate': array(False),
                  'word_splitting': array(False),
                  'punctuation': '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~ ',
                  'lang': 'en',
                  'n_inj': 1,
                  'text_in': 'l£@tspeak',
                  'text_out': 'l£;@tspeak'}},
                'tag': 'MIX',
                'leet_idxs': (22, 32)},
               {'kw': 'text',
                'init_idxs': (32, 36),
                'kw_leet': '₮£><t',
                'params': {'leetspeak': {'change_prb': 0.8,
                  'change_frq': 0.5,
                  'mode': array('covid_intermediate', dtype='<U18'),
                  'get_all_combs': False,
                  'uniform_change': array(False),
                  'seed': 20,
                  'list_changes': [('a', ['@', '4', 'Δ', '*', '', '.']),
                   ('e', ['3', '€', '£', '%', '@', '*', '', '.']),
                   ('i', ['1', 'l', '¡', '!', "'", '*', '', '.']),
                   ('o', ['0', 'ø', '*', '', '.']),
                   ('oo', ['u']),
                   ('u', ['_', 'Ü', 'ü', '*', '.']),
                   ('b', ['ß', 'vb', 'bv']),
                   ('c', ['q', 'k', '©']),
                   ('d', ['t']),
                   ('f', ['ƒ', 'ph']),
                   ('h', ['#']),
                   ('k', ['K']),
                   ('l', ['1', ' l']),
                   ('m', ['ʍ']),
                   ('n', ['π', '¬']),
                   ('p', ['Ᵽ']),
                   ('r', ['₹']),
                   ('s', ['5', '$', 'z']),
                   ('t', ['7', '₮']),
                   ('v', ['b', 'vb', 'bv', '\\/', '▼']),
                   ('w', ['ω']),
                   ('x', ['><', 'kks', '×']),
                   ('y', ['¥']),
                   ('z', ['ƶ'])],
                  'text_in': 'text',
                  'text_out': '₮£><t'}},
                'tag': 'LEETSPEAK',
                'leet_idxs': (33, 38)}]),
             ('leet_sentence',
              'This is an example of l£;@tspeak ₮£><t for NER data generation')])
```

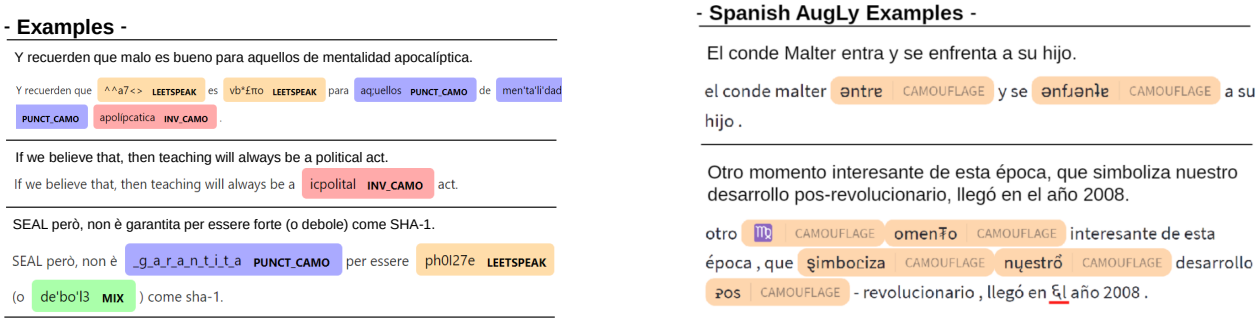
Figure 5.2: Pyleetspeak word camouflage generation outputs metadata detailing the selected words from the original text and the type of word camouflaging applied.

evaluated the usefulness of semantic knowledge for this task, the generalization capabilities and benefits compared to monolingual tasks, and an in-depth evaluation of the types of camouflages where models struggle the most.

5.3.1 Camouflaged Data Simulation

Considering the first contribution of this research, it includes the development of a customizable, multilingual methodology for simulating content evasion. As better explained in the Methodology Section 3.5, drawing from academic literature [233] and observed social media behaviors, the simulation of content evasion was achieved through the development of the "pyleetspeak" Python package, which supports over 20 languages[1] and is available for public use. This package can simulate various text camouflage techniques such as leetspeak, punctuation manipulation, and word inversion but is highly customizable to include new or different techniques. Additionally, the generated data from the "pyleetspeak" tool can be output in various widely used formats such as IOB, JSON, and BILOU format, along with metadata of the modification applied and the positions modified to ease reproducibility and transparency (see Figure 5.2). Examples of the output and capabilities of the tool are shown in Figure 5.3.

As described in Section 3.4, another clear contribution was employing data from diverse sources, including OPUS News-Commentary, OPUS ParaCrawl, TED2020, and WikiMatrix, to generate a synthetic multilingual dataset of camouflaged words annotated with examples of word camouflage in English, Spanish, French, Italian, and German from these datasets. This dataset was made publicly available through the NLP-MisInfo-23 workshop from SEPLN [240].



(a) OPUS ParaCrawl example

(b) Augly Example

Figure 5.3: Multilingual NER model examples in train and external validation data

5.3.2 Multilingual NER Model and Comparative Analysis

The second main contribution comes from using the synthetic camouflaged data to train a multilingual Named Entity Recognition (NER) model for detecting camouflaged content. The study compared various multilingual transformer models to identify the most suitable

[1] ar, az, da, de, el, en, es, fi, fr, hu, id, it, kk, nb, ne, nl, pt, ro, ru, sl, sv, tg, tr

one for detecting word camouflage, evaluating also the impact of semantic knowledge as a pre-training task. Among the models evaluated, the MPNET pre-trained on the STSb dataset developed in the first article (Section 5.1) was included along with the MPNET-base model. The MPNET model pre-trained on the STSb was presented in the IDEAL 2021 congress and will be referred to as MPNET-IDEAL from here on. The reason for comparing different multilingual models lies in their potential to generalize better across multiple languages, a crucial feature for effective content moderation in diverse linguistic environments.

The MPNET model, particularly the MPNET-IDEAL variant, stood out due to its consistent performance and higher accuracy across all tested languages, achieving an overall weighted F1 score of 0.8795. The MPNET-IDEAL model consistently outperformed monolingual models in detecting camouflaged content across multiple languages. While monolingual models achieved their best performance (not better than the multilingual model) just in their respective languages, they lacked the generalizability of multilingual models, struggling with cross-lingual content and varied camouflage techniques.

The multilingual model's ability to generalize across languages was a significant advantage. While monolingual models were effective in their respective languages, they struggled with cross-lingual content, where camouflage techniques and linguistic structures varied significantly.

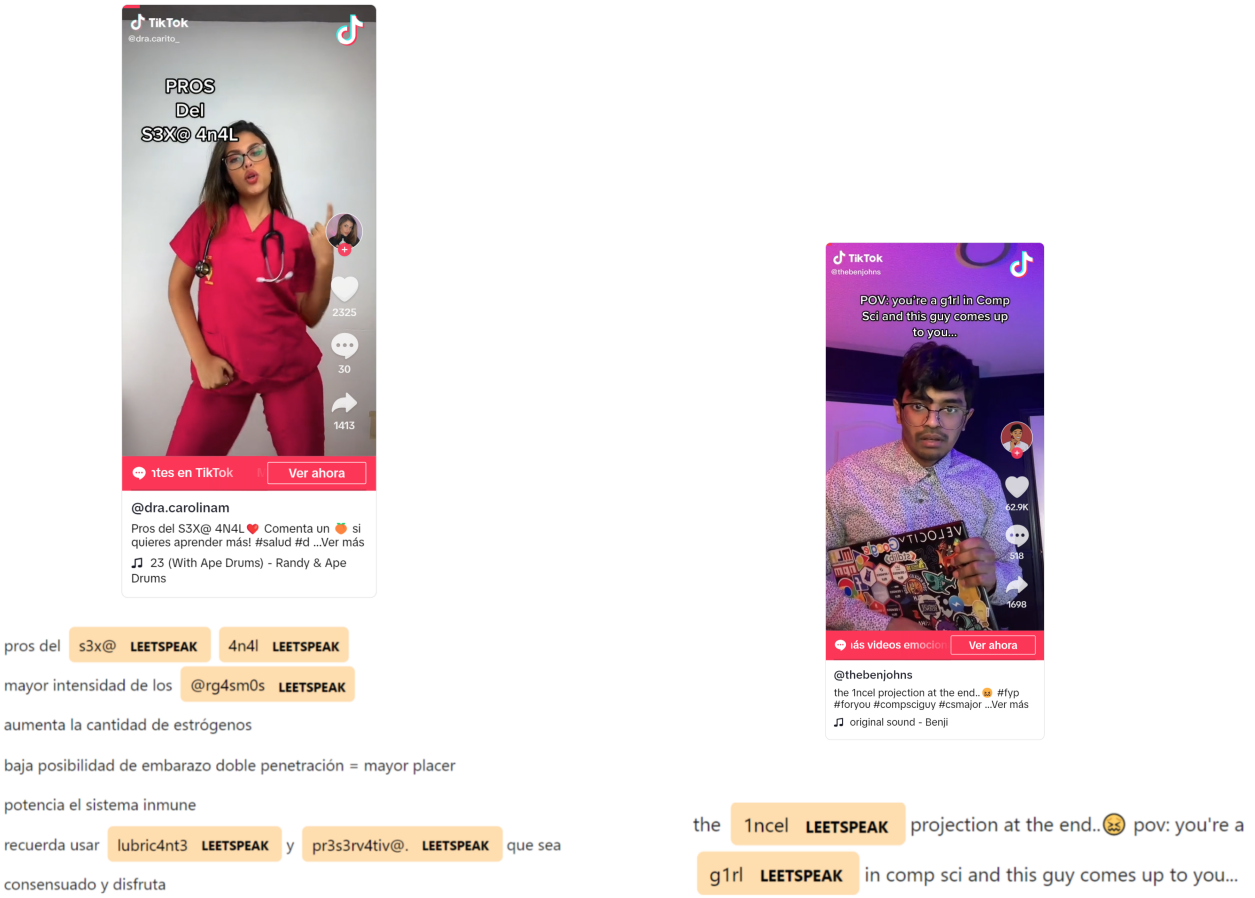
5.3.3 Model Performance and External Validation

The performance was not just evaluated regarding detection scores in terms of classification but also broken down by type of technique applied using a confusion matrix. The confusion matrix analysis of MPNET-IDEAL showed that it excelled in detecting leetspeak and punctuation manipulation but faced challenges with mixed techniques and word inversions. The performance was consistent across different languages and types of resources (types of text with different casual and formal content), showcasing the model's robustness and adaptability. Qualitatively, the confusion matrix analysis provided insights into the model's ability to handle different camouflage techniques. While the model performed well in detecting leetspeak and punctuation manipulation, it faced challenges with mixed techniques and word inversions, suggesting areas for refinement. External validation using Meta AI's AugLy library confirmed the model's effectiveness in detecting novel camouflage strategies, although it struggled with non-keyword modifications.

5.3.4 Impact and Future Directions

This article's contributions align closely with the overall goals of the thesis, focusing on enhancing the robustness of feature spaces in multilingual transformers to combat misinformation. The development of tools like "pyleetspeak" and the multilingual NER model, particularly the MPNET-IDEAL variant, which incorporates semantic knowledge, directly addresses the need for sophisticated techniques to detect and mitigate content evasion, a critical aspect of combating misinformation.

The practical implications of this research are significant for enhancing content moderation systems on social networks. A HuggingFace Space was created, where the MPNET-IDEAL



(a) TikTok Example 1

(b) TikTok Example 2

Figure 5.4: Real-world Examples NER Detection of Camouflaged Words on TikTok (Disclaimer: These examples are shown solely for research purposes and not intended to promote or disseminate any harmful content.

model was made available for testing, further disseminating the research findings[2]. The contributions were also disseminated through various academic forums, including the NLP-MisInfo-23 workshop from SEPLN [240], the COSTAC internal UPM PhD Program workshop, and the CAEPIA Congress 2024. The publicly shared model and dataset enhance research transparency and encourage the replication of results shared at[3].

Future work could explore detecting additional evasion techniques, such as emoticons (emotextuality), typosquatting (URL hijacking), and steganography, to further strengthen content moderation strategies. Investigating the integration of the developed techniques into real-world content moderation systems could also provide valuable insights and practical applications. In fact, the impacts of these techniques on model performance and potential countermeasures are explored in the next article associated with this thesis.

[2] https://twitter.com/spacy_io/status/1483845119879127042

[3] https://codeocean.com/capsule/6133510/tree

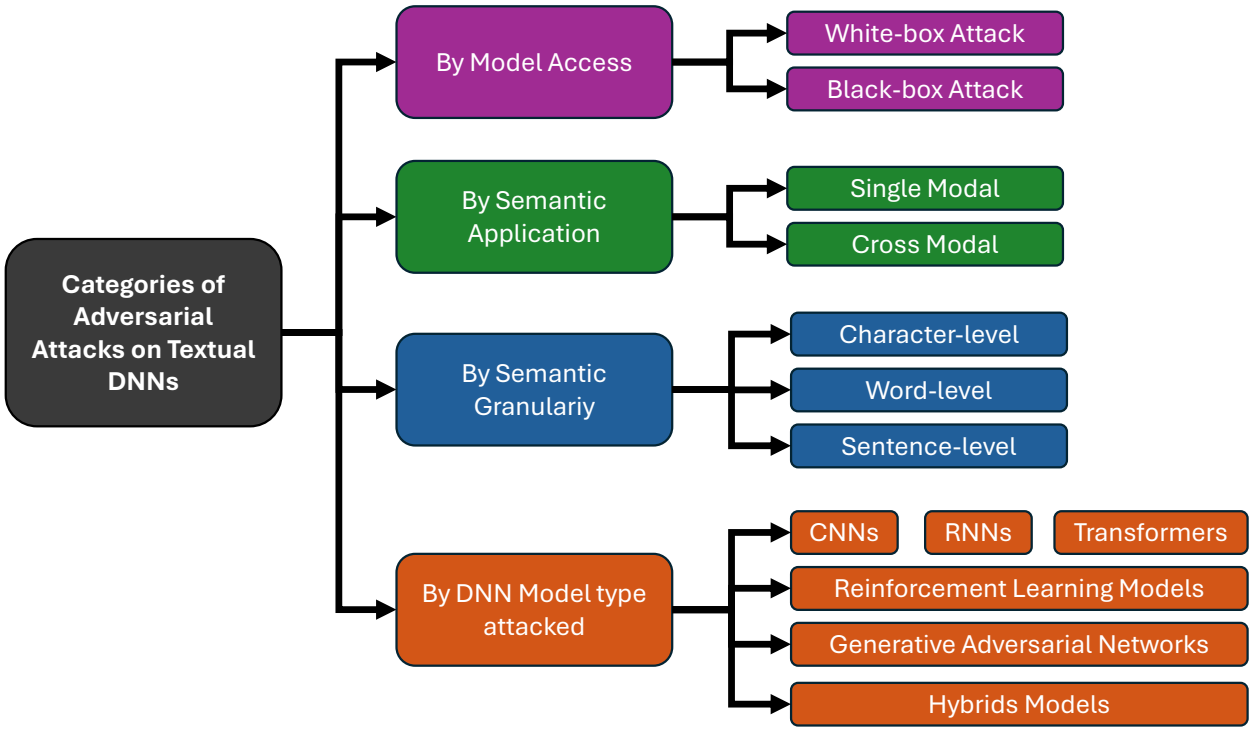


Figure 5.5: Taxonomy of Adversarial methods on textual deep-learning models. Adapted from [238].

5.4 Robustness Against Camouflage adversarial Attack for Content Evasion

Building upon the findings from previous studies, the third article 'Camouflage is all you need: Evaluating and Enhancing Transformer Models Robustness Against Camouflage Adversarial Attacks', addresses the vulnerabilities of advanced language models to word camouflage techniques.

The research focuses on analyzing and enhancing the resilience of Transformer-based NLP models against camouflage adversarial attacks. It investigates the robustness of various Transformer model architectures against manipulated textual inputs designed to evade detection, employing a methodical variation in attack complexity (levels), word camouflage ratio (i.e., number of keywords camouflaged), and instance camouflage ratio (i.e., number of camouflaged instances across the dataset).

The study highlights the critical need for more adaptable tokenizers capable of handling sophisticated adversarial strategies and explores methods to enhance model robustness through adversarial training. These models are tested under progressively challenging conditions to identify vulnerabilities and improve their adversarial robustness.

		Example 1	Example 2	Example 3
	Original	WhoIsQ	WheresTheServer	DumpNike
	Camouflaged	Wh0!sQ	WheresTheServēr	DümpN!kŒ
BERT (WordPiece)	Original Tokens	[who, ##is, ##q]	[where, ##st, ##hes, ##er, ##ver]	[dump, ##nik, ##e]
	Leet Tokens	[w, ##h, ##0, !, sq]	[where, ##st, ##hes, ##er, ##ver]	[dump, ##n, !, k, ##œ]
	Overlapping	0	1	0.33333
mBART (SentencePiece)	Original Tokens	[_Who, Is, Q]	[_Where, s, The, Server]	[_Du, mp, Ni, ke]
	Leet Tokens	[_W, h, 0, !, s, Q]	[_Where, s, The, Ser, vē, r]	[_Dü, mp, N, !, k, Œ]
	Overlapping	0.33333	0.75	0.25
Pythia (BPE)	Original Tokens	[Who, Is, Q]	[W, heres, The, Server]	[D, ump, N, ike]
	Leet Tokens	[Wh, 0, !, s, Q]	[W, heres, The, Serv, Äĵ, r]	[D, Ã¼, mp, N, !, k, Å, Ĵ]
	Overlapping	0.33333	0.75	0.5

Table 5.3: Impact of camouflage on tokenization. Ratio of token overlap before and after camouflaging. values range from 0 (no token overlap) to 1 (identical tokenization), showing the consistency of the tokenization process

Types of Attacks

Adversarial attacks on textual deep neural networks (DNNs) can be categorized based on different criteria, as described in the taxonomy shown in Figure 5.5. Following this classification, the types of attacks explored in the study are:

- By Model Access: The study focuses on black-box attacks, which reflect real-world scenarios where attackers do not have access to model internals.
- By Semantic Application: The research addresses single-modal attacks, targeting text alone without involving other modalities.
- By Semantic Granularity: The study explores character-level, word-level, and to some extent sentence-level modifications, emphasizing the detailed changes that adversarial attacks can introduce.
- By Attacked DNNs: The research primarily focuses on Transformer models, including encoder-only, decoder-only, and encoder-decoder configurations, but is cross-data and cross-model transferable.

5.4.1 Evaluating Transformer Model Vulnerabilities

The study begins by examining the susceptibility of different transformer configurations (encoder-decoder represented by mBART, encoder-only represented by BERT, and decoder-only represented by Pythia) to camouflaged adversarial inputs. To this purpose, two analysis are conduced: the tokenization assesment, and the performance of naive models.

Tokenization

This evaluation examines how tokenization strategies such as WordPiece, BPE, and SentencePiece affect model susceptibility to camouflaged texts, which is crucial for understanding initial model vulnerabilities. The analysis includes an examination of vocabulary sizes and

shared tokens with the camouflage vocabulary. Additionally, as Table 5.3 depicts, the token overlap before and after camouflaging is analyzed with values range from 0 (no token overlap) to 1 (identical tokenization), showing the consistency of the tokenization process under random and keywords camouflage conditions.

The results revealed that regardless of their vocabulary diversity, all tokenizers exhibited notable weaknesses against camouflaged words, particularly in keyword attacks. This vulnerability underscores the need for more adaptable tokenizers capable of handling sophisticated adversarial strategies. Consistent with these findings, transformer models without specific training against word camouflage were increasingly compromised as the complexity and volume of camouflaged inputs grew.

Naive Models

This analysis evaluates the performance of baseline Transformer models (BERT, Pythia, mBART) trained on standard datasets without exposure to any form of camouflage, establishing a performance benchmark. The results demonstrated that all models, regardless of their Transformer architecture, exhibit a notable reduction in performance with increased camouflage complexity. This aligns with the expectation that sophisticated camouflage techniques more severely challenge performance when models are exposed to camouflaged adversarial inputs.

5.4.2 Countermeasures and Adversarial Training

The study further investigates the effectiveness of external preprocessing filters, such as MASK and BLANK, in mitigating the impact of camouflage attacks. The results indicate that external countermeasures enhance naive model performance at lower camouflage levels, particularly when a lower ratio of words is camouflaged. However, these external filters do not improve, and even underperform, the scores of the Naive model at complex levels and with increasing ratios of word camouflage. Interestingly, these results suggest that raw camouflaged text still conveys semantic information that models can utilize. Overall, these external countermeasures provide a comparative standard for model performance with and without these defenses, highlighting the limitations of preprocessing as a sole strategy and paving the way for exploring the intrinsic capabilities of Transformer models to deal with camouflage attacks.

5.4.3 Adversarial Training and Model Robustness

To address the identified vulnerabilities, in the research both static and dynamic adversarial training strategies are applied. Static training involves a one-time introduction of camouflaged data, whereas dynamic training continuously updates the training dataset with new camouflaged examples in each batch during the training process. This dynamic approach not only mitigates the immediate effects of adversarial attacks but also enhances overall model performance by improving the models' ability to generalize from and adapt to evolving adversarial inputs. Additionally, the dynamic approach offers benefits over the static approach, alleviating the learning plateau that static models suffer at high proportions of camouflaged

data instances. Dynamic training shows significant promise, particularly under high camouflage complexities and instance ratios. For instance, models trained with dynamic camouflage exposure exhibit less performance reduction compared to those trained with static approaches. This trend supports the "exposure diversity" benefit of dynamic training, which aligns with the pre-training tasks of models like BERT and RoBERTa, where dynamic masking leads to improved language understanding and generalization.

Moreover, contributions of this research include enhancing the open-source tool pyleetspeak to facilitate the creation of augmented camouflaged datasets, providing researchers and practitioners with effective tools to strengthen NLP systems against continuously evolving threats in digital communication.

By situating this study within the broader narrative of the thesis, which encompasses false information, semantics, and dimensionality reduction, this final research article provides a comprehensive examination of how adversarial attacks can impact and be countered. This focus fills a critical research gap and aligns with global regulatory frameworks like the EU AI Act [244], emphasizing the necessity for robust AI systems capable of maintaining integrity and reliability against adversarial challenges. Through this work, the thesis contributes significantly to the ongoing discourse on ensuring the security and efficacy of AI applications in real-world scenarios.

5.5 Answer the Research Questions

Based on the research conducted in this thesis, the following answers can be provided to the research questions:

5.5.1 Principal Research Questions

- RQ P1: The dimensionality of feature spaces in multilingual transformer models can be optimized through the application of advanced dimensionality reduction techniques, particularly Independent Component Analysis (ICA). This approach reduces the embedding size and maintains semantic integrity across languages. The research demonstrated that ICA can significantly improve model performance while reducing computational overhead, enhancing both efficiency and effectiveness in processing diverse linguistic data.
- RQ P2: To detect and counteract content evasion techniques in multilingual contexts, the thesis developed the “pyleetspeak” tool for simulating and detecting word camouflage. This innovative approach, combined with Name Entity Recognition techniques, was confirmed to be highly effective in identifying camouflaged content within transformer models. The research consistently demonstrated the superiority of multilingual models over monolingual ones in detecting evasion tactics across different languages.
- RQ P3: The understanding of content evasion tactics was integrated into the feature space of NLP models through a combination of external and internal countermeasures. The research explored adversarial training strategies, focusing on in-model awareness

to enhance resilience against sophisticated misinformation techniques. This approach effectively prepared models to combat evolving evasion tactics, significantly improving their robustness and performance in real-world scenarios.

5.5.2 Additional Research Questions

- RQ A1 and RQ A2: The research conclusively demonstrated that the fight against misinformation can be effectively approached from a multilingual perspective. Multilingual models consistently outperformed monolingual models in detecting and mitigating misinformation across various languages. The performance impact of using multilingual strategies was significant, justifying the effort and resources required to extend monolingual datasets to a multilingual level.
- RQ A3: The dimensionality reduction techniques developed for multilingual transformer models showed great promise in their application to a domain-agnostic framework for combating misinformation. The research indicated that these techniques, particularly ICA, can be effectively applied across different domains, enhancing the versatility and applicability of misinformation detection systems.
- RQ A4: The research identified several key bottlenecks and challenges in the ongoing fight against misinformation, including the rapid evolution of evasion techniques, the complexity of processing multilingual data, and the computational demands of large-scale models. From a computational perspective, the thesis proposed solutions such as optimized feature space reduction, efficient multilingual processing, and developing adaptive, semantically-aware models to address these challenges effectively.

5.6 Conclusion and Future Research

This thesis has addressed the critical challenge of combating false information in the digital age through advanced Natural Language Processing (NLP) techniques. As the proliferation of misinformation continues to pose significant threats to public discourse and societal well-being, our research has focused on enhancing the robustness and effectiveness of multilingual Transformer models in detecting and mitigating various forms of false information.

This thesis research has made several significant contributions to the field of NLP and its application in fighting misinformation:

- Optimizing Feature Spaces: We developed innovative techniques to optimize the dimensionality of feature spaces in multilingual Transformer models, enhancing their semantic integrity and performance in processing diverse linguistic data (RQ P1). The optimization of feature spaces in multilingual Transformer models has demonstrated the potential to enhance model performance while reducing computational overhead, making these advanced NLP techniques more accessible and efficient.
- Content Evasion Detection: We introduced novel methods to detect and counteract content evasion techniques in multilingual contexts, including the development of the "pyleetspeak" tool for simulating and detecting word camouflage (RQ P2). The

Figure 5.6: Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis

development of tools like "pyleetspeak" and our multilingual NER model for detecting camouflaged content represents a significant advancement in identifying and countering evolving evasion tactics used by malicious actors.

- Adversarial Training Strategies: We explored the integration of content evasion tactics into the feature space of NLP models and evaluated the effectiveness of adversarial training in preparing these models to combat sophisticated misinformation techniques (RQ P3). The exploration of adversarial training strategies has shown promise in enhancing model robustness against sophisticated misinformation techniques, potentially paving the way for more resilient NLP systems.

As we reflect on the overall impact and challenges of this research, a ”strengths, weaknesses, opportunities, and threats" (SWOT) analysis (Figure 5.6) provides valuable insights. The thesis study's strengths lie in its robust methodology, innovative techniques like the "pyleetspeak" tool, and commitment to open science principles. These strengths have allowed us to make significant advancements in combating misinformation across multiple languages. The research presents opportunities to improve content moderation on a global scale and foster collaboration through open-source tools. However, we must acknowledge weaknesses such as the rapid pace of NLP advancements potentially outpacing our current methods and the limitations of synthetic data in replicating real-world scenarios. Additionally, the focus on NLP approaches, while powerful, does not address all aspects of misinformation, such as the role of visual content or the psychological factors influencing the spread of false information.Threats to consider include the potential for false negatives in the models developed and the continuous

evolution of evasion techniques by malicious actors.

Building on the thesis findings and the ever evolving field of Artificial Intelligence, several recommendation for future research are:

- Further explore the dimensionality reduction techniques, like ICA, alongside the new quantization techniques for faster inference and efficient storage in semantic search applications.
- Exploration of multimodal approaches that combine NLP with visual and audio analysis to create more comprehensive misinformation detection systems.
- Further investigation into the ethical implications of AI-driven content moderation, including strategies to minimize false positives and protect freedom of expression.
- Development of more sophisticated adversarial training techniques that can anticipate and counteract emerging evasion tactics.
- Integration of our NLP models with human-in-the-loop systems to combine the efficiency of AI with human judgment and contextual understanding.
- Expansion of multilingual datasets and models to include a wider range of languages and dialects, particularly those from underrepresented regions.

As we stand at the intersection of artificial intelligence and the fight against misinformation, the research underscores the critical role that advanced NLP techniques can play in fostering a more trustworthy and informed digital landscape. While technology alone cannot solve the complex issue of misinformation, our work contributes to a growing arsenal of tools and strategies that, when combined with critical thinking and media literacy efforts, can help build a more resilient and discerning global community. The battle against misinformation is ongoing, and as researchers, we must remain vigilant, adaptive, and committed to the ethical application of AI in this crucial domain. Our work serves not as an endpoint, but as a foundation and a call to action for continued innovation in the service of truth and informed public discourse.

References

[1] N. Maslej, L. Fattorini, E. Brynjolfsson, *et al.*, *Artificial intelligence index report 2023*, 2023. arXiv: 2310.03715 [cs.AI].

[2] H. Nguyen, D. Nawara, and R. Kashef, "Connecting the indispensable roles of iot and artificial intelligence in smart cities: A survey," *Journal of Information and Intelligence*, 2024, ISSN: 2949-7159. DOI: https://doi.org/10.1016/j.jiixd.2024.01.003.

[3] B. O. Abisoye, Y. Sun, and W. Zenghui, "A survey of artificial intelligence methods for renewable energy forecasting: Methodologies and insights," *Renewable Energy Focus*, vol. 48, p. 100 529, 2024, ISSN: 1755-0084. DOI: https://doi.org/10.1016/j.ref.2023.100529.

[4] S. Polevikov, "Advancing ai in healthcare: A comprehensive review of best practices," *Clinica Chimica Acta*, vol. 548, p. 117 519, 2023, ISSN: 0009-8981. DOI: https://doi.org/10.1016/j.cca.2023.117519.

[5] I. H. Sarker, "AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems," en, *SN Computer Science*, vol. 3, no. 2, p. 158, Mar. 2022, ISSN: 2662-995X, 2661-8907. DOI: 10.1007/s42979-022-01043-x. (visited on 01/21/2024).

[6] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. DOI: 10.48550/ARXIV.1706.03762.

[7] X. Wang, H. Wang, and D. Yang, "Measure and improve robustness in NLP models: A survey," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 4569–4586. DOI: 10.18653/v1/2022.naacl-main.339.

[8] F.-J. Rodrigo-Ginés, J. Carrillo-de-Albornoz, and L. Plaza, "A systematic review on media bias detection: What is media bias, how it is expressed, and how to detect it," *Expert Systems with Applications*, vol. 237, p. 121 641, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121641.

[9] C. M. Greco and A. Tagarelli, "Bringing order into the realm of Transformer-based language models for artificial intelligence and law," en, *Artificial Intelligence and Law*, Nov. 2023, ISSN: 0924-8463, 1572-8382. DOI: 10.1007/s10506-023-09374-7. (visited on 01/21/2024).

[10] B. G. Southwell, E. A. Thorson, and L. Sheble, "The persistence and peril of misinformation: Defining what truth means and deciphering how human brains verify

information are some of the challenges to battling widespread falsehoods," *American Scientist*, vol. 105, pp. 372–375, 2017.

[11] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing &; Management*, vol. 57, no. 2, p. 102 025, Mar. 2020. DOI: 10.1016/j.ipm.2019.03.004.

[12] G. Ruffo, A. Semeraro, A. Giachanou, and P. Rosso, "Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language," *Computer Science Review*, vol. 47, p. 100 531, 2023, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2022.100531.

[13] M. Del Vicario, A. Bessi, F. Zollo, *et al.*, "The spreading of misinformation online," *Proceedings of the National Academy of Sciences*, vol. 113, no. 3, pp. 554–559, Jan. 2016. DOI: 10.1073/pnas.1517441113.

[14] G. K. Shahi, A. Dirkson, and T. A. Majchrzak, "An exploratory study of covid-19 misinformation on twitter," *Online Social Networks and Media*, vol. 22, pp. 100 104–100 104, 2020.

[15] Y. Wang, M. McKee, A. Torbica, and D. Stuckler, "Systematic Literature Review on the Spread of Health-related Misinformation on Social Media," *Social Science & ; Medicine*, vol. 240, p. 112 552, Nov. 2019. DOI: 10.1016/j.socscimed.2019.112552.

[16] H. Allcott, M. Gentzkow, and C. Yu, "Trends in the diffusion of misinformation on social media," *Research &; Politics*, vol. 6, no. 2, p. 205 316 801 984 855, Apr. 2019. DOI: 10.1177/2053168019848554.

[17] W.-Y. S. Chou, A. Oh, and W. M. P. Klein, "Addressing Health-Related Misinformation on Social Media," *JAMA*, vol. 320, no. 23, pp. 2417–2418, Dec. 2018, ISSN: 0098-7484. DOI: 10.1001/jama.2018.16865.

[18] W. H. Organization *et al.*, "Infodemic management: An overview of infodemic management during covid-19, january 2020–may 2021," 2021.

[19] UNESCO and IPSOS, "Survey on the impact of online disinformation and hate speech," en, Tech. Rep., Sep. 2023. [Online]. Available: https://policycommons.net/artifacts/6952470/unesco_ipsos_survey/.

[20] S. Tasnim, M. M. Hossain, and H. Mazumder, "Impact of rumors and misinformation on covid-19 in social media," *Journal of Preventive Medicine and Public Health*, vol. 53, pp. 171–174, 2020.

[21] S. Nieminen and L. Rapeli, "Fighting misperceptions and doubting journalists' objectivity: A review of fact-checking literature," *Political Studies Review*, vol. 17, no. 3, pp. 296–309, 2019. DOI: 10.1177/1478929918786852.

[22] E. K. Vraga and L. Bode, "Using expert sources to correct health misinformation in social media," *Science Communication*, vol. 39, no. 5, pp. 621–645, 2017. DOI: 10.1177/1075547017731776.

[23] R. Salaverría and B. León, "Misinformation beyond the media: 'fake news' in the big data ecosystem," in *Total Journalism: Models, Techniques and Challenges*, J. Vázquez-Herrero, A. Silva-Rodríguez, M.-C. Negreira-Rey, C. Toural-Bran, and X. López-García, Eds. Cham: Springer International Publishing, 2022, pp. 109–121, ISBN: 978-3-030-88028-6. DOI: 10.1007/978-3-030-88028-6_9.

[24] T. Grmuša, "Journalism,„fake news "and disinformation: A handbook for journalism education and training," *Mostariensia: časopis za društvene i humanističke znanosti*, vol. 24, no. 1, pp. 157–159, 2020.

[25] C. Wardle and H. Derakhshan, *Information disorder: Toward an interdisciplinary framework for research and policymaking*, 2017.

[26] N. Karlova and K. E. Fisher, "A social diffusion model of misinformation and disinformation for understanding human information behaviour," *Inf. Res.*, vol. 18, 2013.

[27] H. Situngkir, "Spread of hoax in social media," 2011.

[28] P. Meel and D. K. Vishwakarma, "Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities," *Expert Systems with Applications*, vol. 153, p. 112 986, 2020, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.112986.

[29] M. A. Paz, J. Montero-Díaz, and A. Moreno-Delgado, "Hate speech: A systematized review," *Sage Open*, vol. 10, no. 4, p. 2 158 244 020 973 022, 2020.

[30] P. Tsantarliotis, E. Pitoura, and P. Tsaparas, "Troll vulnerability in online social networks," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2016, pp. 1394–1396.

[31] A. M. Guess and B. A. Lyons, "Misinformation, disinformation, and online propaganda," *Social media and democracy: The state of the field, prospects for reform*, vol. 10, 2020.

[32] X.-r. Zeng, S. Jain, A. Nguyen, and S. Allan, "New perspectives on citizen journalism," *Global Media and China*, vol. 4, pp. 12–3, 2019.

[33] A. Dave, I. V. Chremos, and A. A. Malikopoulos, "Social media and misleading information in a democracy: A mechanism design approach," *IEEE Transactions on Automatic Control*, vol. 67, pp. 2633–2639, 2020.

[34] A. Estrada-Cuzcano, K. Alfaro-Mendives, and V. Saavedra-Vásquez, "Disinformation y misinformation, posverdad y fake news: Precisiones conceptuales, diferencias, similitudes y yuxtaposiciones," *Información, cultura y sociedad*, no. 42, pp. 93–106, 2020.

[35] N. Hassan, B. Adair, J. T. Hamilton, *et al.*, "The quest to automate fact-checking," 2015.

[36] K. Hall, V. Chang, and C. Jayne, "A review on Natural Language Processing Models for COVID-19 research," *Healthcare Analytics*, vol. 2, p. 100 078, 2022, ISSN: 2772-4425. DOI: https://doi.org/10.1016/j.health.2022.100078.

[37] L. Oneto, K. Bunte, and N. Navarin, "Advances in artificial neural networks, machine learning and computational intelligence," *Neurocomputing*, vol. 470, pp. 300–303, 2022, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.07.053.

[38] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, ISSN: 0001-0782. DOI: 10.1145/365153.365168.

[39] M. D. Gordin, *Scientific Babel: how science was done before and after global English*. Chicago ; London: The University of Chicago Press, 2015, ISBN: 9780226000299 9780226000329.

[40] J. Weizenbaum, "ELIZA:a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/365153.365168.

[41] K. Cho, B. van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[42] M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, *et al.*, "Apertium: A free/open-source platform for rule-based machine translation," *Machine translation*, vol. 25, no. 2, pp. 127–144, 2011.

[43] I. Lauriola, A. Lavelli, and F. Aiolli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, 2022, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.05.103.

[44] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51–89, 2005, ISSN: 00664200. DOI: 10.1002/aris.1440370103.

[45] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: An introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011. DOI: 10.1136/amiajnl-2011-000464.

[46] R. Miikkulainen, *Subsymbolic natural language processing: An integrated model of scripts, lexicon, and memory*. MIT press, 1993.

[47] N. Indurkhya and F. J. Damerau, *Handbook of natural language processing*. CRC Press, 2010, vol. 2.

[48] E. Cambria, Y. Li, F. Z. Xing, S. Poria, and K. Kwok, "Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 105–114.

[49] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.

[50] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2017, vol. 1.

[51] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends," *Natural Language Processing Journal*, vol. 4, p. 100 026, 2023, ISSN: 2949-7191. DOI: https://doi.org/10.1016/j.nlp.2023.100026.

[52] C. Toraman, E. Yilmaz, F. Sahinuc, and O. Ozcelik, "Impact of tokenization on language models: An analysis for turkish," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, no. 4, 2023, ISSN: 2375-4699. DOI: 10.1145/3578707.

[53] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162.

[54] J. Song, J.-K. Lee, J. Choi, and I. Kim, "Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences," *Journal of Computational Design and Engineering*, vol. 7, no. 5, pp. 563–576, May 2020, ISSN: 2288-5048. DOI: 10.1093/jcde/qwaa046.

[55] S. S. Stevens, "On the theory of scales of measurement," *Science*, vol. 103, no. 2684, pp. 677–680, 1946. DOI: 10.1126/science.103.2684.677.

[56] C.-F. Tsai, "Bag-of-Words Representation in Image Annotation: A Review," en, *ISRN Artificial Intelligence*, vol. 2012, pp. 1–19, Nov. 2012, ISSN: 2090-7443. DOI: 10.5402/2012/376804. [Online]. Available: https://www.hindawi.com/journals/isrn/2012/376804/ (visited on 02/22/2024).

[57] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003, ISSN: 0306-4573. DOI: https://doi.org/10.1016/S0306-4573(02)00021-3.

[58] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 60, pp. 493–502, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:2996187.

[59] T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola, "Stemming and lemmatization in the clustering of finnish text documents," in *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, ser. CIKM '04, Washington, D.C., USA: Association for Computing Machinery, 2004, pp. 625–633, ISBN: 1581138741. DOI: 10.1145/1031171.1031285.

[60] J. Thanaki, *Python Natural Language Processing: explore NLP with machine learning and deep learning techniques*. Birmingham Mumbai: Packt, 2017, ISBN: 9781787121423.

[61] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[62] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

[63] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Computational Intelligence Magazine*, vol. 13, pp. 55–75, 2017.

[64] K. Babić, S. Martinčić-Ipšić, and A. Meštrović, "Survey of neural text representation models," *Information*, vol. 11, no. 11, 2020, ISSN: 2078-2489. DOI: 10.3390/info11110511.

[65] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[66] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, *Learning word vectors for 157 languages*, 2018. arXiv: 1802.06893 [cs.CL].

[67] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds., Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431.

[68] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527.

[69] D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds., *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*. Cambridge, MA, USA: MIT Press, 1986, ISBN: 026268053X.

[70] K. K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, and D. C. Wunsch II, "4 - selected approaches to supervised learning," in *Computational Learning Approaches to Data Analytics in Biomedical Applications*, K. K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, and D. C. Wunsch II, Eds., Academic Press, 2020, pp. 101–123, ISBN: 978-0-12-814482-4. DOI: https://doi.org/10.1016/B978-0-12-814482-4.00004-8.

[71] R. Sowmyalakshmi, P. M. Venkatesh, T. Jayasankar, and K. Shankar, "Chapter 9 - class imbalance data handling with deep learning–based ubiquitous healthcare monitoring system using wearable devices," in *Wearable Telemedicine Technology for the Healthcare Industry*, H. D. Jude, D. Gupta, A. Khanna, and A. Khamparia, Eds., Academic Press, 2022, pp. 123–136, ISBN: 978-0-323-85854-0. DOI: https://doi.org/10.1016/B978-0-323-85854-0.00010-1.

[72] G. E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002, ISSN: 0899-7667. DOI: 10.1162/089976602760128018.

[73] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," en, *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539.

[74] W. Yin, K. Kann, M. Yu, and H. Schütze, *Comparative study of cnn and rnn for natural language processing*, 2017. arXiv: 1702.01923 [cs.CL].

[75] A. Kedia and M. Rasu, *Hands-On Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications*, eng. Birmingham Mumbai: Packt, 2020, ISBN: 9781838989590.

[76] T. Zheng, Y. Gao, F. Wang, *et al.*, "Detection of medical text semantic similarity based on convolutional neural network," en, *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, p. 156, Dec. 2019, ISSN: 1472-6947. DOI: 10.1186/s12911-019-0880-2. (visited on 04/01/2024).

[77] S. Sengupta, S. Basak, P. Saikia, *et al.*, "A review of deep learning with special emphasis on architectures, applications and recent trends," *Knowl. Based Syst.*, vol. 194, p. 105 596, 2019.

[78] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, vol. 1, 1996, 347–352 vol.1. DOI: 10.1109/ICNN.1996.548916.

[79] A. Madsen, "Visualizing memorization in RNNs," *Distill*, vol. 4, no. 3, 10.23915/distill.00016, Mar. 2019, ISSN: 2476-0757. DOI: 10.23915/distill.00016.

[80] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[81] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, D. Wu, M. Carpuat, X. Carreras, and E. M. Vecchi, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012.

[82] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2016. arXiv: 1409.0473 [cs.CL].

[83] K. Xu, J. L. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15, Lille, France: JMLR.org, 2015, pp. 2048–2057.

[84] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022, ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2022.10.001.

[85] S. Islam, H. Elmekki, A. Elsebai, *et al.*, "A comprehensive survey on applications of transformers for deep learning tasks," *Expert Systems with Applications*, vol. 241, p. 122 666, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122666.

[86] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. DOI: 10.18653/v1/P19-1285.

[87] N. Adaloglou and S. Karagiannakos, "How attention works in deep learning: Understanding the attention mechanism in sequence models," *https://theaisummer.com/*, 2020. [Online]. Available: https://theaisummer.com/attention/.

[88] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," vol. 55, no. 6, Dec. 2022, ISSN: 0360-0300. DOI: 10.1145/3530811.

[89] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[90] J. L. Ba, J. R. Kiros, and G. E. Hinton, *Layer normalization*, 2016. arXiv: 1607.06450 [stat.ML].

[91] K. S. Kalyan, "A survey of gpt-3 family large language models including chatgpt and gpt-4," *Natural Language Processing Journal*, vol. 6, p. 100 048, 2024, ISSN: 2949-7191. DOI: https://doi.org/10.1016/j.nlp.2023.100048.

[92] J. Ding, S. Ma, L. Dong, *et al.*, "Longnet: Scaling transformers to 1,000,000,000 tokens," in *Proceedings of the 10th International Conference on Learning Representations*, 2023.

[93] B. Peng, S. Narayanan, and C. Papadimitriou, *On limitations of the transformer architecture*, 2024. arXiv: 2402.08164 [stat.ML].

[94] Q. Wen, T. Zhou, C. Zhang, *et al.*, "Transformers in time series: A survey," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, E. Elkind, Ed., Survey Track, International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 6778–6786. DOI: 10.24963/ijcai.2023/759.

[95] S. Luo, S. Li, T. Cai, *et al.*, *Stable, fast and accurate: Kernelized attention with relative positional encoding*, 2021. arXiv: 2106.12566 [cs.LG].

[96] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: The long-document transformer*, 2020. arXiv: 2004.05150 [cs.CL].

[97] B. Chen, T. Dao, E. Winsor, Z. Song, A. Rudra, and C. Ré, *Scatterbrain: Unifying sparse and low-rank attention approximation*, 2021. arXiv: 2110.15343 [cs.LG].

[98] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, *Linformer: Self-attention with linear complexity*, 2020. arXiv: 2006.04768 [cs.LG].

[99] A. Jaegle, S. Borgeaud, J.-B. Alayrac, *et al.*, *Perceiver io: A general architecture for structured inputs & outputs*, 2022. arXiv: 2107.14795 [cs.LG].

[100] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," *Advances in neural information processing systems*, vol. 28, 2015.

[101] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031.

[102] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410.

[103] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 4512–4525. DOI: 10.18653/v1/2020.emnlp-main.365.

[104] Y. Bondarenko, M. Nagel, and T. Blankevoort, "Understanding and overcoming the challenges of efficient transformer quantization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7947–7969. DOI: 10.18653/v1/2021.emnlp-main.627.

[105] J. Li, T. Zhang, I. E.-H. Yen, and D. Xu, *Fp8-bert: Post-training quantization for transformer*, 2023. arXiv: 2312.05725 [cs.AI].

[106] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 10 088–10 115. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

[107] H. Wang, S. Ma, L. Dong, *et al.*, *Bitnet: Scaling 1-bit transformers for large language models*, 2023. arXiv: 2310.11453 [cs.CL].

[108] S. Ma, H. Wang, L. Ma, *et al.*, *The era of 1-bit llms: All large language models are in 1.58 bits*, 2024. arXiv: 2402.17764 [cs.CL].

[109] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127 063, 2024.

[110] OpenAI, J. Achiam, S. Adler, *et al.*, *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL].

[111] G. Team, R. Anil, S. Borgeaud, *et al.*, *Gemini: A family of highly capable multimodal models*, 2023. arXiv: 2312.11805 [cs.CL].

[112] H. Touvron, T. Lavril, G. Izacard, *et al.*, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2302.13971.

[113] T. Wolf, L. Debut, V. Sanh, *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Q. Liu and D. Schlangen, Eds., Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[114] S. Bhattamishra, A. Patel, and N. Goyal, *On the computational power of transformers and its implications in sequence modeling*, 2020. arXiv: 2006.09286 [cs.LG].

[115] Instituto Cervantes, Ed., *El español en el mundo 2023: anuario del Instituto Cervantes*. Madrid: Instituto Cervantes : Bala Perdida Editorial, 2023.

[116] S. Kemp, *Digital 2024: Global Overview Report*, en-GB, publisher: DataReportal, Jan. 2024. [Online]. Available: https://datareportal.com/reports/digital-2024-global-overview-report (visited on 03/30/2024).

[117] R. Wang and H. Zhao, "Advances and challenges in unsupervised neural machine translation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, I. Augenstein and I. Habernal, Eds., online: Association for Computational Linguistics, Apr. 2021, pp. 17–21. DOI: 10.18653/v1/2021.eacl-tutorials.5.

[118] C. Si, Z. Zhang, Y. Chen, *et al.*, "Sub-Character Tokenization for Chinese Pretrained Language Models," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 469–487, May 2023, ISSN: 2307-387X. DOI: 10.1162/tacl_a_00560.

[119] A. Metke-Jimenez., K. Raymond., and I. MacColl., "Information extraction from web services - a comparison of tokenisation algorithms," in *Proceedings of the 2nd International Workshop on Software Knowledge (IC3K 2011) - SKY*, INSTICC, SciTePress, 2011, pp. 12–23, ISBN: 978-989-8425-82-9. DOI: 10.5220/0003698000120023.

[120] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, *Ammus : A survey of transformer-based pretrained models in natural language processing*, 2021. arXiv: 2108.05542 [cs.CL].

[121] J. Yang, *Rethinking tokenization: Crafting better tokenizers for large language models*, 2024. arXiv: 2403.00417 [cs.CL].

[122] S. Kemp, *Digital 2022: Language, Culture, and Global Content Habits*, en-GB, Jan. 2022. [Online]. Available: https://datareportal.com/reports/digital-2022-language-culture-and-content (visited on 03/30/2024).

[123] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, "The state and fate of linguistic diversity and inclusion in the NLP world," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 6282–6293. DOI: 10.18653/v1/2020.acl-main.560.

[124] A. CONNEAU and G. Lample, "Cross-lingual language model pretraining," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

[125] J. J. Webster and C. Kit, "Tokenization as the initial phase in nlp," in *Proceedings of the 14th Conference on Computational Linguistics - Volume 4*, ser. COLING '92, Nantes, France: Association for Computational Linguistics, 1992, pp. 1106–1110. DOI: 10.3115/992424.992434.

[126] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16, Phoenix, Arizona: AAAI Press, 2016, pp. 2741–2749.

[127] X. Zhang and Y. LeCun, *Text understanding from scratch*, 2016. arXiv: 1502.01710 [cs.LG].

[128] A. Rai and S. Borah, "Study of various methods for tokenization," in *Applications of Internet of Things*, J. K. Mandal, S. Mukhopadhyay, and A. Roy, Eds., Singapore: Springer Singapore, 2021, pp. 193–200, ISBN: 978-981-15-6198-6.

[129] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5149–5152. DOI: 10.1109/ICASSP.2012.6289079.

[130] P. Gage, "A new algorithm for data compression," *C Users J.*, vol. 12, no. 2, pp. 23–38, Feb. 1994, ISSN: 0898-9788.

[131] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:160025533.

[132] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[133] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[134] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, E. Blanco and W. Lu, Eds., Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012.

[135] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 66–75. DOI: 10.18653/v1/P18-1007.

[136] T. Limisiewicz, J. Balhar, and D. Mareček, *Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages*, 2023. arXiv: 2305.17179 [cs.CL].

[137] M. Lewis, Y. Liu, N. Goyal, *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703.

[138] L. Xue, N. Constant, A. Roberts, *et al.*, "MT5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41.

[139] M. Kale, A. Siddhant, N. Constant, M. Johnson, R. Al-Rfou, and L. Xue, *Nmt5 – is parallel data still relevant for pre-training massively multilingual language models?* 2021. arXiv: 2106.02171 [cs.CL].

[140] P.-J. Chen, J. Shen, M. Le, *et al.*, "Facebook AI's WAT19 Myanmar-English translation task submission," in *Proceedings of the 6th Workshop on Asian Translation*, T. Nakazawa, C. Ding, R. Dabre, *et al.*, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 112–122. DOI: 10.18653/v1/D19-5213.

[141] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747.

[142] I. Doval, S. F. Lanza, T. J. Juliá, E. Liste Lamas, and B. Lübke, "Corpus PaGeS: A multifunctional resource for language learning, translation and cross-linguistic research," en, in *Studies in Corpus Linguistics*, I. Doval and M. T. Sánchez Nieto, Eds., vol. 90, Amsterdam: John Benjamins Publishing Company, Mar. 2019, pp. 103–121, ISBN: 9789027202345 9789027262844. DOI: 10.1075/scl.90.07dov. [Online]. Available: https://benjamins.com/catalog/scl.90.07dov (visited on 03/31/2024).

[143] J. Tiedemann, "News from opus - a collection of multilingual parallel corpora with tools and interfaces," odefinierat/okänt, in *Recent Advances in Natural Language Processing*, N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, Eds. 2009, vol. V, pp. 237–248.

[144] L. D. Liello, M. Gabburo, and A. Moschitti, *Efficient pre-training objectives for transformers*, 2021. arXiv: 2104.09694 [cs.CL].

[145] L. Di Liello, M. Gabburo, and A. Moschitti, "Effective pretraining objectives for transformer-based autoencoders," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5533–5547. DOI: 10.18653/v1/2022.findings-emnlp.405.

[146] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020. [Online]. Available: https://openreview.net/pdf?id=r1xMH1BtvB.

[147] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245.

[148] H. Fang, S. Wang, M. Zhou, J. Ding, and P. Xie, *Cert: Contrastive self-supervised learning for language understanding*, 2020. arXiv: 2005.12766 [cs.CL].

[149] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV].

[150] Y. Liu, J. Gu, N. Goyal, *et al.*, "Multilingual Denoising Pre-training for Neural Machine Translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, Nov. 2020, ISSN: 2307-387X. DOI: 10.1162/tacl_a_00343. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00343/1923401/tacl_a_00343.pdf.

[151] Y. Tang, C. Tran, X. Li, *et al.*, "Multilingual translation from denoising pre-training," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 3450–3466. DOI: 10.18653/v1/2021.findings-acl.304.

[152] X. Jiang, Y. Liang, W. Chen, and N. Duan, "Xlm-k: Improving cross-lingual language model pre-training with multilingual knowledge," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 10 840–10 848, Jun. 2022. DOI: 10.1609/aaai.v36i10.21330.

[153] J. Phang, T. Févry, and S. R. Bowman, *Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks*, 2019. arXiv: 1811.01088 [cs.CL].

[154] Y. Zhou and V. Srikumar, *A closer look at how fine-tuning changes bert*, 2022. arXiv: 2106.14282 [cs.CL].

[155] M. Mosbach, A. Khokhlova, M. A. Hedderich, and D. Klakow, "On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers," in *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, A. Alishahi, Y. Belinkov, G. Chrupała, D. Hupkes, Y. Pinter, and H. Sajjad, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 68–82. DOI: 10.18653/v1/2020.blackboxnlp-1.7.

[156] H. Touvron, L. Martin, K. Stone, *et al.*, *Llama 2: Open foundation and fine-tuned chat models*, 2023. arXiv: 2307.09288 [cs.CL].

[157] A. Conneau, R. Rinott, G. Lample, *et al.*, "XNLI: Evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2475–2485. DOI: 10.18653/v1/D18-1269.

[158] M. Artetxe, S. Ruder, and D. Yogatama, "On the cross-lingual transferability of monolingual representations," *CoRR*, vol. abs/1910.11856, 2019. arXiv: 1910.11856.

[159] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, *Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization*, 2020. arXiv: 2003.11080 [cs.CL].

[160] P. Lewis, B. Oguz, R. Rinott, S. Riedel, and H. Schwenk, "MLQA: Evaluating cross-lingual extractive question answering," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter,

and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7315–7330. DOI: 10.18653/v1/2020.acl-main.653.

[161] S. Ruder, *The State of Multilingual AI*, http://ruder.io/state-of-multilingual-ai/, 2022.

[162] T. Marwala, E. Fournier-Tombs, and S. Stinckwich, *The use of synthetic data to train ai models: Opportunities and risks for sustainable development*, 2023. arXiv: 2309.00652 [cs.LG].

[163] A. F. Aji, G. I. Winata, F. Koto, *et al.*, "One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7226–7249. DOI: 10.18653/v1/2022.acl-long.500.

[164] G. Kuwanto, A. F. Akyürek, I. C. Tourni, S. Li, A. G. Jones, and D. Wijaya, *Low-resource machine translation training curriculum fit for low-resource languages*, 2021. arXiv: 2103.13272 [cs.CL].

[165] P. Liang, R. Bommasani, T. Lee, *et al.*, *Holistic evaluation of language models*, 2023. arXiv: 2211.09110 [cs.CL].

[166] L. Zhou and D. Zhang, "An ontology-supported misinformation model: Toward a digital misinformation library," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 5, pp. 804–813, 2007. DOI: 10.1109/TSMCA.2007.902648.

[167] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, R. Barzilay and M. Johnson, Eds., Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 1589–1599. [Online]. Available: https://aclanthology.org/D11-1147.

[168] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11, Hyderabad, India: Association for Computing Machinery, 2011, pp. 675–684, ISBN: 9781450306324. DOI: 10.1145/1963405.1963500.

[169] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17, Singapore, Singapore: Association for Computing Machinery, 2017, pp. 797–806, ISBN: 9781450349185. DOI: 10.1145/3132847.3132877.

[170] Y. Liu, M. Ott, N. Goyal, *et al.*, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL].

[171] X. Zeng, A. S. Abumansour, and A. Zubiaga, "Automated fact-checking: A survey," en, *Language and Linguistics Compass*, vol. 15, no. 10, e12438, Oct. 2021, ISSN: 1749-818X, 1749-818X. DOI: 10.1111/lnc3.12438. (visited on 04/02/2024).

[172] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, Sep. 2020, ISSN: 0360-0300. DOI: 10.1145/3395046.

[173] D. Q. Nguyen, T. Vu, and A. Tuan Nguyen, "BERTweet: A pre-trained language model for English tweets," in *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing: System Demonstrations, Q. Liu and D. Schlangen, Eds., Online: Association for Computational Linguistics, Oct. 2020, pp. 9–14. DOI: 10.18653/v1/2020.emnlp-demos.2.

[174] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke, and L. Neves, "TweetEval: Unified benchmark and comparative evaluation for tweet classification," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 1644–1650. DOI: 10.18653/v1/2020.findings-emnlp.148.

[175] J. Zhou, J. Tian, R. Wang, Y. Wu, W. Xiao, and L. He, "SentiX: A sentiment-aware pre-trained model for cross-domain sentiment analysis," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds., Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 568–579. DOI: 10.18653/v1/2020.coling-main.49.

[176] Y.-F. Huang and P.-H. Chen, "Fake news detection using an ensemble learning model based on self-adaptive harmony search algorithms," *Expert Systems with Applications*, vol. 159, p. 113 584, 2020, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113584.

[177] S. Tufchi, A. Yadav, and T. Ahmed, "A comprehensive survey of multimodal fake news detection techniques: Advances, challenges, and opportunities," en, *International Journal of Multimedia Information Retrieval*, vol. 12, no. 2, p. 28, Dec. 2023, ISSN: 2192-6611, 2192-662X. DOI: 10.1007/s13735-023-00296-3. (visited on 04/04/2024).

[178] A. Giachanou, G. Zhang, and P. Rosso, "Multimodal Fake News Detection with Textual, Visual and Semantic Information," en, in *Text, Speech, and Dialogue*, P. Sojka, I. Kopeček, K. Pala, and A. Horák, Eds., vol. 12284, Cham: Springer International Publishing, 2020, pp. 30–38, ISBN: 9783030583224 9783030583231. DOI: 10.1007/978-3-030-58323-1_3.

[179] S. Shaar, N. Babulkov, G. Da San Martino, and P. Nakov, "That is a known lie: Detecting previously fact-checked claims," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 3607–3618. DOI: 10.18653/v1/2020.acl-main.332.

[180] Á. Huertas-García, H. Liz, G. Villar-Rodríguez, A. Martín, J. Huertas-Tato, and D. Camacho, "AIDA-UPM at SemEval-2022 task 5: Exploring multimodal late information fusion for multimedia automatic misogyny identification," in *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, G. Emerson, N. Schluter, G. Stanovsky, *et al.*, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 771–779. DOI: 10.18653/v1/2022.semeval-1.107.

[181] Á. Huertas-García, A. Martín, J. Huertas-Tato, and D. Camacho, "Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflage," *Applied Soft Computing*, vol. 145, p. 110 552, 2023, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2023.110552.

[182] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[183] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[184] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[185] A. Moi and N. Patry, *Huggingface's tokenizers*, version 0.13.4, Apr. 2023. [Online]. Available: https://github.com/huggingface/tokenizers.

[186] I. Montani, M. Honnibal, M. Honnibal, A. Boyd, S. V. Landeghem, and H. Peters, *explosion/spaCy: v3.7.2: Fixes for APIs and requirements*, version v3.7.2, Oct. 2023. DOI: 10.5281/zenodo.10009823.

[187] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[188] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018. DOI: 10.21105/joss.00861.

[189] github, *Github*, 2020. [Online]. Available: https://github.com/.

[190] Z. Papakipos and J. Bitton, *Augly: Data augmentations for robustness*, 2022. arXiv: 2201.06494 [cs.AI].

[191] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "YAKE! Keyword extraction from single documents using multiple local features," en, *Information Sciences*, vol. 509, pp. 257–289, Jan. 2020, ISSN: 00200255. DOI: 10.1016/j.ins.2019.09.013.

[192] M. Grootendorst, *Keybert: Minimal keyword extraction with bert.* Version v0.1.3, 2020. DOI: 10.5281/zenodo.4461265.

[193] S. Bird and E. Loper, "NLTK: The natural language toolkit," in *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 214–217. [Online]. Available: https://aclanthology.org/P04-3031.

[194] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, "Cupy: A numpy-compatible library for nvidia gpu calculations," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. [Online]. Available: http://learningsys.org/nips17/assets/papers/paper_16.pdf.

[195] P. T. Inc. "Collaborative data science." (2015), [Online]. Available: https://plot.ly.

[196] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[197] M. L. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. DOI: 10.21105/joss.03021.

[198] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," en, in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, Philadelphia, PA, USA: ACM Press, 2006, p. 535, ISBN: 9781595933393. DOI: 10.1145/1150402.1150464.

[199] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015. arXiv: 1503.02531 [stat.ML].

[200] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, *Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring*, 2020. arXiv: 1905.01969 [cs.CL].

[201] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*, 2020. arXiv: 1910.01108 [cs.CL].

[202] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, *Language-agnostic bert sentence embedding*, 2020. arXiv: 2007.01852 [cs.CL].

[203] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, *Mpnet: Masked and permuted pre-training for language understanding*, 2020. arXiv: 2004.09297 [cs.CL].

[204] Á. Huertas-García, J. Huertas-Tato, A. Martín García, and D. Camacho, "Countering Misinformation Through Semantic-Aware Multilingual Models," in *Intelligent Data Engineering and Automated Learning – IDEAL 2021*, Springer International Publishing, 2021, pp. 312–323, ISBN: 978-3-030-91607-7. DOI: 10.1007/978-3-030-91608-4_31.

[205] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. DOI: 10.18653/v1/S17-2001.

[206] N. Muennighoff, T. Wang, L. Sutawika, *et al.*, *Crosslingual generalization through multitask finetuning*, 2022. DOI: 10.48550/ARXIV.2211.01786.

[207] T. L. Scao, A. Fan, C. Akiki, *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint*, 2022. DOI: arXiv:2211.05100.

[208] L. Xue, N. Constant, A. Roberts, *et al.*, *Mt5: A massively multilingual pre-trained text-to-text transformer*, 2020. DOI: 10.48550/ARXIV.2010.11934.

[209] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, "Unsupervised cross-lingual representation learning at scale," *CoRR*, vol. abs/1911.02116, 2019. arXiv: 1911.02116.

[210] S. Biderman, H. Schoelkopf, Q. Anthony, *et al.*, *Pythia: A suite for analyzing large language models across training and scaling*, 2023. arXiv: 2304.01373 [cs.CL].

[211] Y. Tang, C. Tran, X. Li, *et al.*, "Multilingual translation with extensible multilingual pretraining and finetuning," 2020. arXiv: 2008.00401 [cs.CL].

[212] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

[213] L. Li, K. Jamieson, A. Rostamizadeh, *et al.*, *Massively parallel hyperparameter tuning*, 2018. [Online]. Available: https://openreview.net/forum?id=S1Y7OOlRZ.

[214] L. Liu, X. Liu, J. Gao, W. Chen, and J. Han, "Understanding the difficulty of training transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 5747–5763. DOI: 10.18653/v1/2020.emnlp-main.463.

[215] S. Singh and A. Mahmood, "The nlp cookbook: Modern recipes for transformer based deep learning architectures," *IEEE Access*, vol. 9, pp. 68 675–68 702, 2021, ISSN: 2169-3536. DOI: 10.1109/access.2021.3077350.

[216] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, 2019. arXiv: 1711.05101 [cs.LG].

[217] B. Zhuang, J. Liu, Z. Pan, H. He, Y. Weng, and C. Shen, *A survey on efficient training of transformers*, 2023. arXiv: 2302.01107 [cs.LG].

[218] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. DOI: 10.18653/v1/S17-2001.

[219] Á. Huertas-García, J. Huertas-Tato, A. Martín, and D. Camacho, "Countering misinformation through semantic-aware multilingual models," in *Intelligent Data Engineering and Automated Learning – IDEAL 2021*, H. Yin, D. Camacho, P. Tino, *et al.*, Eds., Cham: Springer International Publishing, 2021, pp. 312–323, ISBN: 978-3-030-91608-4.

[220] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 2214–2218.

[221] M. Bañón, P. Chen, B. Haddow, *et al.*, "ParaCrawl: Web-scale acquisition of parallel corpora," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 4555–4567. DOI: 10.18653/v1/2020.acl-main.417.

[222] H. Schwenk, V. Chaudhary, S. Sun, H. Gong, and F. Guzmán, *Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia*, 2019. arXiv: 1907.05791 [cs.CL].

[223] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 75–86. DOI: 10.18653/v1/S19-2010.

[224] Á. Huertas-García, A. Martín, J. Huertas-Tato, and D. Camacho, "Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflage," *Applied Soft Computing*, vol. 145, p. 110 552, 2023, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2023.110552.

[225] V. E. Shcherbina, L. A. Pasechnaya, O. P. Simutova, I. V. Verzhinskaya, and N. A. Belova, "The Lexicon Of The Gaming Community Members As A Youth Slang Component," Nov. 2022, pp. 678–686. DOI: 10.15405/epsbs.2022.11.92.

[226] M. Kavanagh, "Bridge the generation gap by decoding leetspeak," English, *Inside the Internet*, vol. 12, no. 12, p. 11, 2005.

[227] L. Shifman, "An anatomy of a YouTube meme," en, *New Media & Society*, vol. 14, no. 2, pp. 187–203, Mar. 2012, ISSN: 1461-4448, 1461-7315. DOI: 10.1177/1461444811412160.

[228] J. M. Tellería Gelabert, "English and leetspeak: A step towards global nerdism?," 2012.

[229] J. Fuchs, " Gamespeak for n00bs - a linguistic and pragmatic analysis of gamers' language," English, Ph.D. dissertation, University of Graz, 2013. [Online]. Available: https://unipub.uni-graz.at/obvugrhs/content/titleinfo/231890?lang=en.

[230] R. Craenen, *Leet speak cheat sheet*, en. [Online]. Available: https://www.gamehouse.com/blog/leet-speak-cheat-sheet/ (visited on 01/24/2022).

[231] H.-L. Titangos, *Local community in the era of social media technologies: A global approach*. Elsevier, 2013.

[232] H. Weijo, J. Hietanen, and P. Mattila, "New insights into online consumption communities and netnography," en, *Journal of Business Research*, vol. 67, no. 10, pp. 2072–2078, Oct. 2014, ISSN: 01482963. DOI: 10.1016/j.jbusres.2014.04.015.

[233] A. Romero-Vicente, *Word camouflage to evade content moderation*, en-US, 2021. [Online]. Available: https://www.disinfo.eu/publications/word-camouflage-to-evade-content-moderation/ (visited on 01/23/2022).

[234] M. Delkic, "Leg Booty? Panoramic? Seggs? How TikTok Is Changing Language," en-US, *The New York Times*, Nov. 2022, ISSN: 0362-4331. [Online]. Available: https://www.nytimes.com/2022/11/19/style/tiktok-avoid-moderators-words.html (visited on 04/24/2024).

[235] T. Lorenz, "Internet 'algospeak' is changing our language in real time, from 'nip nops' to 'le dollar bean'," en-US, *Washington Post*, Apr. 2022, ISSN: 0190-8286. [Online]. Available: https://www.washingtonpost.com/technology/2022/04/08/algospeak-tiktok-le-dollar-bean/ (visited on 04/24/2024).

[236] D. Klug, E. Steen, and K. Yurechko, "How algorithm awareness impacts algospeak use on tiktok," in *Companion Proceedings of the ACM Web Conference 2023*, ser. WWW '23 Companion, , Austin, TX, USA, Association for Computing Machinery, 2023, pp. 234–237, ISBN: 9781450394192. DOI: 10.1145/3543873.3587355.

[237] V. Vera, "Nonsuicidal self-injury and content moderation on tiktok," *Proceedings of the Association for Information Science and Technology*, vol. 60, no. 1, pp. 1164–1166, 2023. DOI: 10.1002/pra2.979.

[238] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, 2020, ISSN: 2157-6904. DOI: 10.1145/3374217.

[239] W. Wang, J.-t. Huang, W. Wu, *et al.*, "MTTM: Metamorphic Testing for Textual Content Moderation Software," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, Melbourne, Australia: IEEE, May 2023, pp. 2387–2399, ISBN: 9781665457019. DOI: 10.1109/ICSE48619.2023.00200.

[240] Á. Huertas-García, A. Martín, J. Huertas-Tato, and D. Camacho, "Countering malicious content moderation evasion in online social networks: Simulation and detection of word camouflages," in *Proceedings of the NLP-MisInfo Workshop co-located with 39th International Conference of SEPLN*, ser. CEUR Workshop Proceedings, vol. 3525, CEUR-WS.org, 2023, pp. 10–14. [Online]. Available: https://ceur-ws.org/Vol-3525/paper1.pdf.

[241] Á. Huertas-García, A. Martín, J. Huertas-Tato, and D. Camacho, "Exploring Dimensionality Reduction Techniques in Multilingual Transformers," *Cognitive Computation*, vol. 15, no. 2, pp. 590–612, Mar. 2023, ISSN: 1866-9956, 1866-9964. DOI: 10.1007/s12559-022-10066-8.

[242] Á. Huertas-García, A. Martín, J. Huertas-Tato, and D. Camacho, "Camouflage is all you need: Evaluating and enhancing transformer models robustness against camouflage adversarial attacks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–13, 2024. DOI: 10.1109/TETCI.2024.3440181.

[243] A. Martín, J. Huertas-Tato, Á. Huertas-García, G. Villar-Rodríguez, and D. Camacho, "Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference," *Knowledge-Based Systems*, vol. 251, p. 109 265, 2022, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2022.109265.

[244] *Proposal for a Regulation laying down harmonised rules on artificial intelligence | Shaping Europe's digital future*, en, Apr. 2021. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence.