
Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**


Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

# Uso de Interlinguas para Búsqueda Documental Multilingüe


Autor(a): César García Cabeza
Tutor(a): Jesús Cardeñosa Lera



Madrid, Julio 2021

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

*Trabajo Fin de Máster*
*Máster Universitario en* Inteligencia Artificial

*Título:* Uso de Interlinguas para Búsqueda Documental Multilingüe

Julio 2021

*Autor(a):* César García Cabeza
*Tutor(a):* Jesús Cardeñosa Lera
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

# Agradecimientos

A Jesús, al que considero mucho más que un simple tutor del TFM. Gracias por haberme dado la oportunidad de formar parte de DAIL y de realizar este bonito proyecto.

A Gaba, que a pesar de las circunstancias especiales de no presencialidad del máster, considero un amigo y ha hecho amenos muchos momentos del curso. Tenemos pendiente tomarnos algo algún día.

Por último, pero no por ello menos importante, a mi familia, por haber sido siempre un apoyo constante en cada paso que he dado. Esto es vuestro.

Gracias de corazón.

# Resumen

La búsqueda documental es una de las tareas que se llevan a cabo en el área de la Recuperación de la Información. Esta tarea permite a las personas buscar documentos concretos de entre una colección de estos de manera automática. Un paso más allá es el permitir buscar documentos desde varias lenguas, lo que se conoce por búsqueda documental multilingüe.

En este trabajo se realiza un análisis de los métodos previos desarrollados para solucionar el problema de la multilingualidad en los buscadores documentales. Se propone un modelo basándose en técnicas previas pero con la novedad de usar la interlingua *Universal Networking Language* como puente entre las lenguas.

Finalmente, se implementa dicho modelo a pequeña escala. La experimentación demuestra que, siempre y cuando la calidad de los diccionarios sea buena, el modelo propuesto obtiene buenos resultados al realizar búsquedas desde el inglés al español.

# Abstract

Document search is one of the tasks carried out in the area of Information Retrieval. This task allows people to automatically search for specific documents from a collection of documents. A further step is to allow people to search for documents from several languages, known as multilingual document search.

This work provides an analysis of previous methods developed to solve the problem of multilinguality in document search engines. A model is proposed based on previous techniques but with the novelty of using the interlingua *Universal Networking Language* as a bridge between languages.

Finally, the model is implemented on a small scale. Experimentation shows that, as long as the quality of the dictionaries is good, the proposed model obtains good results when searching from English to Spanish.

# Tabla de contenidos

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

La multilingualidad es hoy en día una característica más del mundo globalizado en el que vivimos. Es común que las personas hablen varias lenguas y que personas de una cultura se comuniquen con personas de otra cultura a través de un medio común, el lenguaje. Esta diferencia entre las lenguas que hablan las personas no puede suponer un factor discriminante a la hora de realizar tareas propias del mundo actual, como el uso de los ordenadores, etc.

Una de estas tareas es la búsqueda de documentos, perteneciente al área de la Recuperación de la Información, que consiste en recuperar documentos que tengan que ver con las necesidades de los usuarios, expresadas por medio de *queries* o consultas. Esta tarea ha sido ampliamente investigada en los últimos años ya que es uno de los pilares fundamentales de las búsquedas en Internet. Sin embargo, el estudio de sistemas multilingües no ha sido tan exhausto y completo como el de los sistemas monolingües.

Desarrollar un sistema de búsqueda documental multilingüe no es una tarea sencilla ya que hay que superar la barrera del lenguaje para así ser capaces de realizar búsquedas multilingües. El principal problema es que las técnicas exploradas hasta la fecha tienen ciertas desventajas, siendo la principal que se enfocan en ambientes bilingües más que en ambientes verdaderamente multilingües y únicamente en pares de lenguas comunes como pueden ser el español, inglés, francés, alemán, italiano o chino.

## 1.1. Objetivos

En esta tesis, uno de los objetivos será conocer la situación actual de los buscadores documentales multilingües y más en detalle su evolución a lo largo de los últimos años.

Tras un profundo estudio, se propondrá un modelo de búsqueda documental multilingüe apoyado en el concepto de ”interlingua” para convertir las palabras de una lengua a otra, de manera que sea sencilla y factible la escalabilidad del sistema en ambientes verdaderamente multilingües. Finalmente, se implementará un sistema basado en el modelo propuesto anteriormente.

## 1.2. Estructura

Este trabajo está estructurado de la siguiente manera:

- En el Capítulo 2 se desarrollará el Estado del Arte de los buscadores documentales multilingües, centrándonos especialmente en las diferentes técnicas de traducción.

- Tras haber explicado el Estado del Arte y los diferentes enfoques para afrontar nuestro problema, en los Capítulos 3 y 4 se planteará el problema de este trabajo junto a una serie de limitaciones o hipótesis de trabajo.
- En el Capítulo 5 se propondrá el modelo de buscador documental multilingüe basado en una interlingua.
- Tras la propuesta de modelo, en el Capítulo 6 se explicará el diseño de experimentación así como la implementación del modelo llevada a cabo y se expondrán los resultados obtenidos.
- Finalmente, en el Capítulo 7 se presentarán las conclusiones finales del trabajo y las posibles líneas futuras de trabajo.

# Capítulo 2

# Estado del Arte

La Búsqueda Documental Multilingüe se trata de una subárea de la Recuperación de la Información (de ahora en adelante RI), llamada en inglés *Information Retrieval*. Por lo tanto, antes de entrar en detalle en la definición de Búsqueda Documental Multilingüe se deberá comprender qué es la RI y qué es la Búsqueda Documental.

La RI es un área de las Ciencias de la Computación encargada principalmente de proporcionar a los usuarios un acceso fácil a la información. De acuerdo con Gerald Salton [50], uno de los pioneros en este campo, una posible definición formal es:

> "La Recuperación de la Información es un campo encargado de la estructura, análisis, organización, almacenamiento, búsqueda y recuperación de la información."

Para ello, los usuarios deberán expresar mediante una consulta, o *query*, la información que desean buscar. Una vez creada esta *query*, la meta principal de un sistema de RI es la de recuperar todos los documentos relevantes para la *query* del usuario a la vez que recuperar el menor número posible de documentos no relevantes [4]. Se considera que un documento es relevante a una *query* del usuario si contiene información relacionada con el contenido de esta; sin embargo, la relevancia no es una característica objetiva y el análisis de la relevancia de los documentos debe ser realizado *ad-hoc* para cada caso concreto.

La Búsqueda Documental es un caso de aplicación concreto dentro de la RI, en la que los usuarios desean recuperar información de una colección cerrada de documentos almacenados. Otro ejemplo de aplicación concreta de la RI serían los motores de búsqueda, estando estos especializados en buscar y recuperar información de la Web sobre un conjunto abierto de documentos, es decir, que cambia con el tiempo. Entendiendo la ínfima diferencia entre RI y Búsqueda Documental, es común en la literatura hacer uso de ambos términos indistintamente.

Una vez definida qué es la RI y la Búsqueda Documental, se puede definir qué es la Búsqueda Documental Multilingüe. Existen varias definiciones de este término, de entre las que se pueden destacar [25]:

1. RI en una colección de documentos paralelos o en una colección multilingüe de documentos, donde el espacio de búsqueda está restringido a la lengua en la que se formula la *query*.
2. RI en una colección monolingüe de documentos, pudiendo realizar *queries* en varias lenguas.
3. RI en una colección multilingüe de documentos, pudiendo realizar *queries* en cualquiera de la lenguas de los documentos.

4. RI en una colección de documentos multilingües, es decir, documentos que contienen información en más de una lengua.

En la definición 1 se trabaja con una colección de documentos en múltiples lenguas pero, los documentos en cada lengua son vistos como colecciones independientes en la que la lengua de la *query* determina la lengua de la búsqueda. Un ejemplo de esta definición sería el motor de búsqueda Google, en el que podemos buscar páginas webs en varias lenguas pero la lengua en la que hacemos la búsqueda determina la de los resultados. Si buscamos páginas webs relacionadas con fútbol, únicamente nos aparecerán como resultado páginas en español relacionadas con fútbol. En el caso de que quisiésemos buscar páginas en inglés relacionadas con fútbol, deberíamos introducir en la búsqueda la palabra "*football*". Las definiciones 2-4 ya se corresponderían con el sistema que planteamos en este trabajo, permitiendo al usuario superar las barreras del lenguaje a la hora de buscar información.

Para evitar la confusión que genera el término "multilingüe", Ballesteros y Croft [5] adoptan el término *Cross-Lingual Information Retrieval* (de ahora en adelante CLIR) para hacer referencia a las definiciones 2-4. Por lo tanto, a partir de ahora se utilizará el término CLIR para referirnos a la capacidad de realizar *queries* en una lengua y buscar o recuperar información (o documentos) en otra lengua.

En el término *cross-lingual* es donde se encuentra uno de los principales problemas de la literatura hasta el momento, y es que acaban reduciendo la búsqueda multilingüe a una mera búsqueda bilingüe. Un ejemplo de un proyecto que, por el contrario, se mantiene fiel a la palabra multilingüe sería HEREIN [23]. Este proyecto europeo surgió de la necesidad de establecer un marco común entre los diferentes países miembro sobre términos de patrimonio cultural, dando lugar a la creación de un tesauro multilingüe en 15 lenguas distintas.

Está claro que para lograr traspasar las barreras del lenguaje a la hora de buscar documentos, se deberá de alguna forma traducir bien sean los documentos o las *queries*. Este hecho añade un gran y nuevo problema a los sistemas tradicionales de búsqueda documental, haciendo que la principal línea de investigación en este campo sea el cómo realizar esta traducción. Una vez que se consiga esta traducción, el problema quedará reducido a una simple búsqueda documental monolingüe.

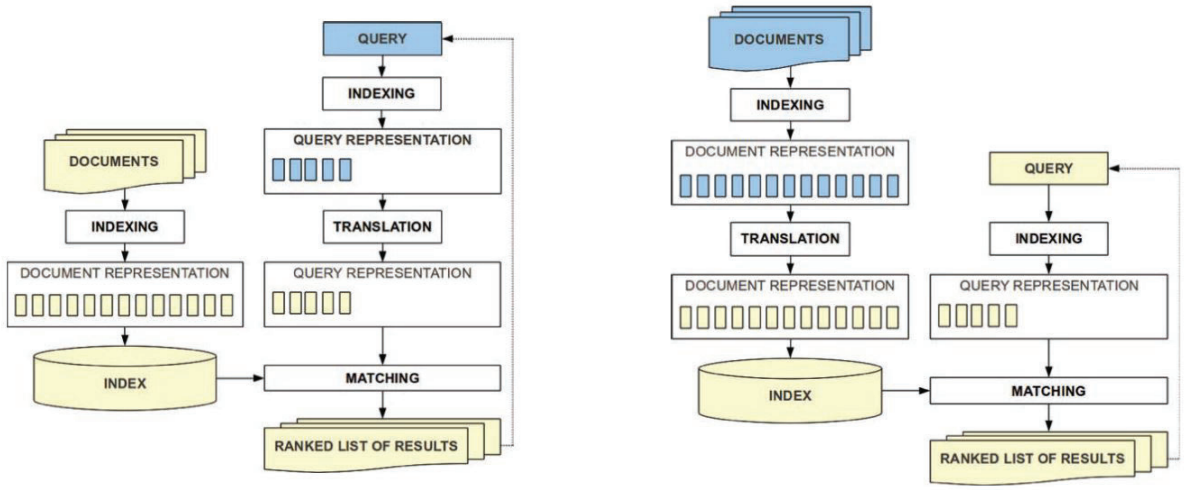
## 2.1. Traducción de la *Query* vs. traducción del Documento

Una vez que sabemos que para lograr buscar información en una lengua haciendo consultas en otra lengua es necesaria una traducción, la primera pregunta que nos surge podría ser: “¿Es mejor traducir la *query* en el momento de la búsqueda o es mejor traducir todos los documentos al indexarlos[1]?”. En la Figura 2.1 se puede ver un ejemplo de cómo sería el esquema general de un sistema CLIR dependiendo de qué se traduzca: las *queries* o los documentos.

La mayor parte de la literatura que se encuentra está centrada en la traducción de la *query*. Las principales ventajas y desventajas de este enfoque son [28], [39]:

- **Ventajas**:
  - **Coste computacional**: el coste computacional de traducir una *query* es mucho menor al de traducir un documento debido a su tamaño.
  - **Almacenamiento**: traducir la *query* no supone duplicar los archivos de indexado y por tanto, aumentar el almacenamiento del sistema. Únicamente se necesita de la incorpo-

[1] Indexar los documentos es el proceso por el cual se representa y almacena la información relevante de un documentos de manera digital.




(a) Basado en traducción de la *query*.

(b) Basado en traducción del documento.

Figura 2.1: Esquema de un sistema CLIR. Fuente: [63]

ración de un modulo de traducción en el momento de realizar la búsqueda (véase Figura 2.1a).

- **Desventajas**:
  - **Ambigüedad**: unos de los principales y mayores problemas de la traducción es la ambigüedad introducida en este proceso. En el caso de traducir la *query*, esta ambigüedad es aún mayor debido a la naturaleza de las *queries*: suelen ser cortas y formadas por palabras sueltas. Este hecho hace que sea más difícil aplicar técnicas de desambiguación lingüística (de ahora en adelante WSD, del inglés *Word-Sense Disambiguation*).

Por otra parte, hay muy poca literatura centrada en la traducción del documento.

- Oard y Hacket [40] realizaron experimentos traduciendo los documentos de alemán a inglés utilizando un sistema de Traducción Automática (de ahora en adelante MT, del inglés *Machine Translation*). Se dieron cuenta que al utilizar técnicas de MT para traducir los documentos, la calidad de la traducción mejoraba debido a que estas se benefician de textos largos para poder desambiguar. No solo mejoraba la calidad de la traducción, sino que además mejoraba la precisión a la hora de recuperar los documentos.
- McCarley [35] realizó experimentos en inglés y francés comparando la eficacia a la hora de recuperar los documentos de tres sistemas: uno que traducía los documentos, otro que traducía las *queries* y otro que traducía ambos. En sus experimentos comprobó que el mejor tipo de sistema de traducción dependía del sentido de la traducción en sí: al traducir de inglés a francés se obtenían mejores resultados traduciendo los documentos y de francés a inglés traduciendo las *queries*. Además,

Por lo que, a modo de resumen, las ventajas y desventajas de traducir los documentos son precisamente las opuestas de traducir las *queries*:

- **Ventajas**:
  - **Ambigüedad**: al tratarse de textos más largos, la ambigüedad que se introduce con la traducción disminuye al poder aplicar técnicas de WSD. En especial, los sistemas de MT obtienen traducciones de muy buena calidad.

  - **Rendimiento**: si acotamos rendimiento al momento de la búsqueda, estos sistemas son más rápidos al no necesitar de una traducción de la *query*, ya que los documentos han sido previamente traducidos.

- **Desventajas**:

  - **Almacenamiento**: como comentamos en las ventajas de traducir las *queries*, traducir los documentos supone almacenar los documentos indexados en varios idiomas (véase Figura 2.1b). Esto supone aumentar el almacenamiento del sistema, dejando de ser escalable en el caso de querer poder buscar documentos entre muchas lenguas.

Debido a lo expuesto anteriormente, prácticamente toda la literatura sobre sistemas CLIR está centrada en la traducción de la *query*.

## 2.2. Arquitectura general de un sistema CLIR

Haciendo una analogía con los sistemas software, se podría hablar de dividir las funcionalidades de un sistema CLIR en módulos más pequeños e independientes, de manera que cada modulo esté especializado en una tarea concreta.

Siguiendo la arquitectura propuesta en [63], se podría hablar de 4 módulos perfectamente diferenciados y conectados entre sí secuencialmente:

- Módulo de pre-traducción
- Módulo de traducción
- Módulo de post-traducción
- Módulo de RI

En la Figura 2.2 se puede observar de manera simplificada la descomposición del sistema CLIR en los 4 módulos descritos. A continuación, se irán describiendo en detalle los módulos mencionado uno a uno.

### 2.2.1. Módulo de pre-traducción

El módulo de pre-traducción es el encargado de preprocesar el texto permitiendo extraer la información importante de este y de preparar la *query* para garantizar la mejor traducción posible. Dentro de este módulo se podrían encontrar las siguientes tareas:

- **Tokenización**: el primer preprocesado en cualquier aplicación de PLN es el separar la información del texto en palabras, a este proceso se le conoce como tokenización.

  Para lenguas como el español, este proceso es de lo más sencillo debido a la naturaleza propia del lenguaje. Por regla general las palabras van separas en el texto por medio de espacios en blanco, haciendo la tokenización una tarea trivial. Por ejemplo, el resultado de tokenizar la frase “Soy un estudiante de la Universidad Politécnica de Madrid” sería:

  Soy, un, estudiante, de, la, Universidad, Politécnica, de, Madrid

  El siguiente nivel en cuanto a dificultad, sería en lenguas como el inglés o el francés donde es frecuente el uso de contracciones. En estos casos es necesario preprocesar estas palabras para pasar de las formas contraídas a las formas canónicas. De esta forma, se conseguiría pasar de la expresión contraída “He’s a student at the Technical University of Madrd” a:

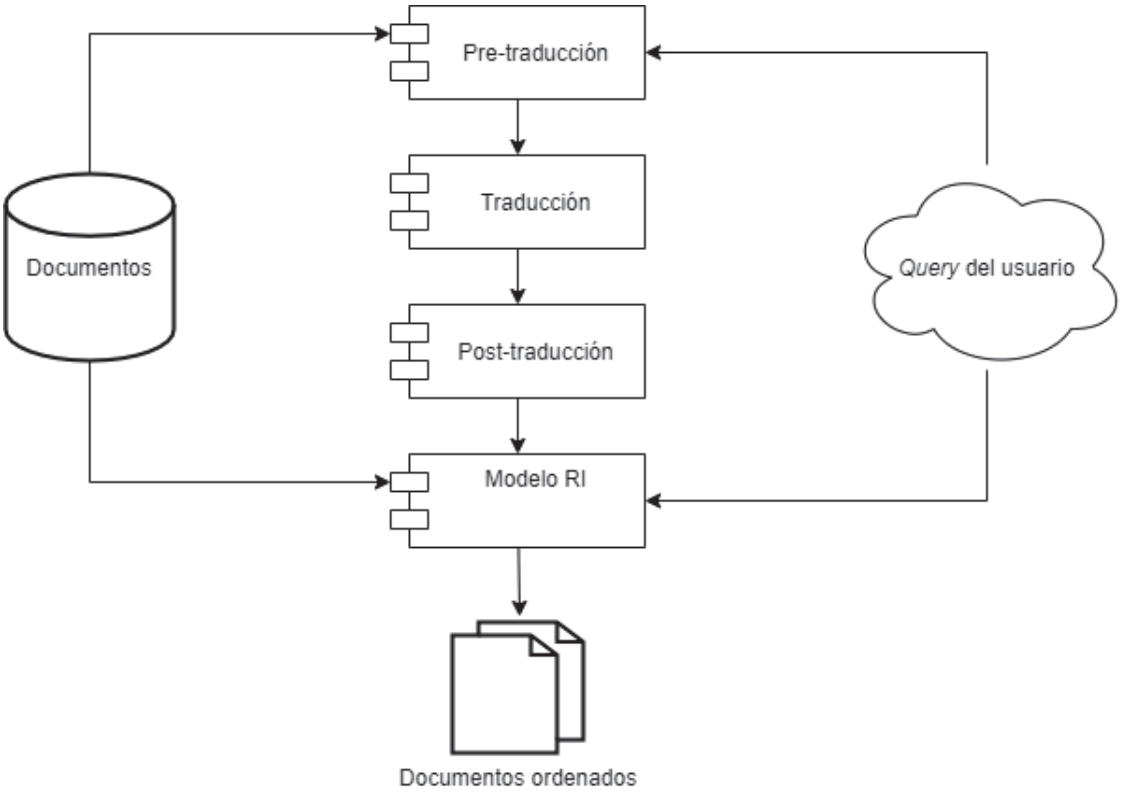



Figura 2.2: Arquitectura común de un sistema CLIR propuesta en [63].

He, is, a, student, at, the, Technical, University, of, Madrid

En lenguas como el chino, el nivel de dificultad es aún mayor debido a que la separación entre palabras no está marcada obligatoriamente por los espacios en blanco. Este hecho hace que la tokenización sea mucho más difícil y sea necesario el uso de técnicas más complejas como la segmentación de palabras.

Por último, es importante ser capaces de detectar las expresiones de más de una palabra en el texto. En un buscador documental monolingüe este hecho no es tan esencial, pero en un buscador documental multilingüe el detectar expresiones influye de manera directa en la traducción y por tanto, en la búsqueda de documentos [5]. Un ejemplo muy sencillo sería la expresión en francés *"pommes de terre"* que significa "patatas" en español. Si no se detectase la expresión, su traducción sería "manzanas de tierra".

- **Eliminación de *stopwords***: la eliminación de las palabras vacías consiste en eliminar del texto todas las palabras que no aportan un significado relevante. Comúnmente se suelen eliminar los artículos, preposiciones, conjunciones, verbos muy usados como "haber" o "ser"... Si eliminamos las palabras vacías del primer ejemplo nos quedaría como resultado:

  estudiante, universidad, Politécnica, Madrid

- **Normalización**: es el proceso encargado de unificar en una única forma términos con significados iguales o definir una equivalencia entre ellos. En el caso de los sistemas CLIR, esta normalización se realiza enfocada a mejorar el *matching* a la hora de traducir [43].

  El primer tipo de normalización posible sería pasar el texto a minúsculas. A pesar de ser muy común esta normalización en cualquier tarea de PLN monolingüe, en el caso de las aplicacio-

nes multilingüe el pasar un nombre propio a minúsculas puede provocar que se traduzca de forma errónea.

El segundo tipo de normalización sería la lematización. Este proceso morfológico consiste en reducir las palabras a sus lemas.

$$\text{jugaré, jugaremos, jugaste} \implies \text{jugar}$$

El último tipo de normalización es el *stemming*, que a diferencia de la lematización no reduce las palabras a su lema sino a su raíz.

$$\text{gato, gata, gatos, gatitos} \implies \text{gat}$$

En [30] se exponen una serie de diferencias entre la lematización y el stemming que hay que tener en cuenta a la hora de normalizar las palabras:

1. Al ser un proceso basado en la morfología, la lematización preserva la categoría morfológica (en inglés *Part of Speech)* de las palabras. Por el contrario, el *stemming* reduce las palabras a su raíz sin tener en cuenta su categoría morfológica.

2. Además, aplicar *stemming* no tiene por qué dar como resultado una palabra existente (como vimos en el anterior ejemplo de los gatos).

- **Expansión de términos**: el último posible preprocesado antes de la traducción sería expandir los términos con otros relacionados. En el caso de la expansión de términos en la pre-traducción, el máximo objetivo es expandir el texto con palabras que ayuden a desambiguar la traducción. Por tanto, realizando una expansión de términos en la pre-traducción se aumenta la precisión de un sistema CLIR [6]. A la hora de realizar la expansión de términos en este módulo, se usan recursos en la misma lengua que los documentos o las *queries* que se van a traducir.

  En la Sección 2.3 se verán tres técnicas de expansión diferentes usadas en la literatura para los sistemas CLIR.

### 2.2.2. Módulo de traducción

El módulo de traducción es el núcleo central de los sistemas CLIR que traducen los documentos o las *queries*, siendo lo que lo diferencia de un sistema monolingüe tradicional. Este módulo es el encargado de traducir la *query* en el momento de búsqueda a la lengua en la que están indexados los documentos o de traducir los documentos a la lengua en la que se van a realizar las consultas en el momento de indexarlos.

A la hora de traducir, se puede hablar de dos tipos bien diferenciados de técnicas: basadas en el conocimiento y basadas en corpus.

- La primera técnica hace uso de alguna fuente de conocimiento, p. ej. diccionarios o tesauros, para realizar la traducción. El principal problema de esta es la ambigüedad introducida en la traducción al tener que escoger el mejor significado según el contexto en el que se encuentre la palabra. Otra desventaja es la limitada existencia de tesauros que sirvan como apoyo en un sistema CLIR.

- La segunda técnica hace uso de corpus[2] multilingües para extraer, mediante métricas estadísticas, similitudes entre términos de distintas lenguas y así poder realizar la traducción. En este

[2] Por corpus entendemos un conjunto organizado de textos que son usados, en este caso, para nuestra tarea de Búsqueda Documental.

caso, el mayor problema es el encontrar corpus multilingües de un tamaño suficiente como para poder crear las similitudes entre los términos de ambas lenguas.

Por otra parte, también hay otras técnicas que no encajarían en ninguna de las dos descritas previamente: basadas en traducción automática, basadas en una interlingua y basadas en aprendizaje profundo.

En la Sección 2.4 se entrará en detalle en la evolución de estas técnicas y se analizarán sus desventajas y ventajas.

### 2.2.3. Módulo de post-traducción

El módulo de post-traducción es el paso previo al módulo de Recuperación de la Información. En este módulo, la única tarea que se lleva a cabo es de nuevo la expansión de términos; pero, en este caso, la expansión se realiza sobre el resultado de la traducción. Al realizar este método se consigue aumentar tanto la cobertura como la precisión [6].

Por cobertura entendemos la proporción de documentos relevantes que un sistema es capaz de devolver de entre el total de documentos relevantes y por precisión la proporción de documentos relevantes devueltos de entre el total de devueltos. Para facilitar la comprensión de estos dos términos, se presenta en la Tabla 2.1 un ejemplo muy sencillo en el que en la primera columna tenemos el ranking de documentos, en la segunda columna si el documento era o no relevante, la tercera columna representa el número de documentos devueltos en cada paso, la cuarta columna significa el número de documentos relevantes devueltos en cada paso y las últimas dos columnas el nivel de cobertura y precisión en cada paso.

Para calcular la cobertura en cada fila se deberá dividir el número de documentos relevantes devueltos entre el total de relevantes (3 en el ejemplo). Para calcular la precisión, se deberá dividir el número de documentos relevantes devueltos entre el número de documentos devueltos.

Tabla 2.1: Ejemplo sobre la precisión y la cobertura.

| Ranking | Relevante | # devueltos | # relevantes devueltos | Cobertura (r=3) | Precisión |
|---|---|---|---|---|---|
| 1 | Sí | 1 | 1 | 0.33 | 1 |
| 2 | No | 2 | 1 | 0.5 | 0.5 |
| 3 | Sí | 3 | 2 | 0.66 | 0.66 |
| 4 | Sí | 4 | 3 | 1 | 0.75 |
| 5 | No | 5 | 3 | 1 | 0.6 |

La expansión de términos en la post-traducción es una técnica muy utilizada en sistemas CLIR que traducen la *query*. Sin embargo, no es tan común en aquellos sistemas que traducen los documentos [63].

### 2.2.4. Módulo de búsqueda

El módulo de búsqueda es el encargado en sí de realizar la búsqueda documental y devolver los documentos más relevantes para la *query* del usuario. Esto se consigue por medio de un modelo que calcula la similitud entre la *query* y cada documento y devuelve los documentos ordenados de mayor a menos similitud. Este modelo es independiente al hecho de que la búsqueda sea monolingüe o multilingüe, simplemente aporta las herramientas para realizar la búsqueda.

En [4] se define un modelo de RI como una cuádrupla [**D**, **Q**, $F$, $R(q_i, d_j)$] donde:

1. **D** es un conjunto formado por las representaciones de los documentos de una colección.
2. **Q** es un conjunto formado por las representaciones de las nececesidades de información de los usuarios (o *queries*).
3. $F$ es un entorno de trabajo capaz de modelar las representaciones de los documentos y de las *queries* y sus relaciones, p. ej. conjuntos y relaciones booleanas o vectores.
4. Finalmente, $R(q_i, d_j)$ es una función de ranking que asocia un número real para la representación de la *query* $q_i \in \mathbf{Q}$ y la representación del documento $d_j \in \mathbf{D}$ en función de su similitud. Este ranking define una ordenación entre los documentos respecto a la *query* $q_i$.

Hablando sobre los modelos de RI, se puede hablar de dos agrupaciones diferenciadas en base a su antigüedad: los modelos clásicos y los modelos modernos.

#### 2.2.4.1. Modelos clásicos

Los modelos clásicos en la RI son el Modelo Booleano, el Modelo del Espacio Vectorial y los Modelos Probabilísticos.

**Modelo Booleano** El Modelo Booleano es considerado como el primer modelo de RI y se basa en la teoría de conjuntos y en el álgebra booleano. Es un modelo muy simple e intuitivo en el que el cálculo de las similitudes entre la *query* y los documentos se hace en base a las operaciones booleanas AND, OR y NOT. Las representaciones de los términos en los documentos y en las *queries* se realizan por medio de frecuencias booleanas, es decir, presente o no presente.

Imaginemos que tenemos un sistema muy simple como el que vemos en la Tabla 2.2, en el que las filas representan el vocabulario de la colección y las columnas los documentos de la colección. Cada posición de la matriz contiene un 1 si el término está presente en el documento o un 0 si no lo está. De esta forma, podemos ver por ejemplo que el documento $D_1$ contiene las palabras *inteligencia* y *artificial*. Por otro lado, una posible *query* sería "*visión* OR *aprendizaje*" y se obtendrían como resultado los documentos $D_2$, $D_3$ y $D_4$ al ser el resultado de realizar una operación lógica OR entre los dos términos.

Tabla 2.2: Representación de los términos y los documentos en el Modelo Booleano.

| | **D1** | **D2** | **D3** | **D4** |
|---|---|---|---|---|
| **inteligencia** | 1 | 0 | 0 | 1 |
| **artificial** | 1 | 0 | 0 | 1 |
| **visión** | 0 | 1 | 1 | 1 |
| **computador** | 0 | 1 | 1 | 1 |
| **aprendizaje** | 0 | 0 | 1 | 1 |

Como resultado de este modelo, la función de ranking genera como salida una relevancia binaria: 1 si el documento es relevante para la *query* o 0 si no lo es. A simple vista se puede ver que este hecho restringe la posibilidad de similitudes o *matchings* parciales. Esto provoca que el tamaño de la salida del modelo sea o muy grande o muy pequeña.

**Modelo del Espacio Vectorial** El Modelo del Espacio Vectorial (o *Vector Space Model*, VSM, en inglés) se presenta en [48] como una solución a la imposibilidad de *matchings* parciales del Modelo Booleano. Para ello, representa los documentos y las consultas de los usuarios en forma de vectores de pesos. Sin embargo, sigue presentando una serie de desventajas [4], [29]:

- Los vectores son dispersos, es decir, la mayoría de posiciones son 0 por lo que supone un almacenamiento extra innecesario.
- Asume que los términos son mutuamente independientes entre ellos.
- No captura la posición (sintaxis) ni el significado (semántica) de las palabras en el texto.
- Si no se normaliza la métrica, la longitud del documento afectará a los valores.

El esquema de ponderación de términos más común es la métrica TF-IDF[3], que mezcla en un solo valor lo frecuente que es un término con su *rareza*. Esta métrica incluye por una parte la frecuencia de un término en el documento o *query* y por otra la frecuencia inversa del documento, es decir, la inversa del número de documentos en los que aparece un término. Existen muchas variaciones de esta métrica, siendo la más común [4]:

$$(1 + log\ f_{i,j}) \times \log \frac{N}{n_i} \tag{2.1}$$

donde $N$ es el número total de documentos, $n_i$ el número de documentos en los que aparece el término i-ésimo y $f_{i,j}$ el número de veces que aparece el término i-ésimo en el documento j-ésimo.

Al representar los documentos y las consultas como vectores, una forma muy sencilla de calcular similitudes es haciendo uso del coseno entre los vectores, también llamada *cosine similarity*. Tenemos un vector $\vec{q}$ en formato tf-idf que representa a la *query*; por otro lado, tenemos un vector $\vec{d}$ en formato tf-idf que representa al documento. Este esquema se puede ver en la Figura 2.3.

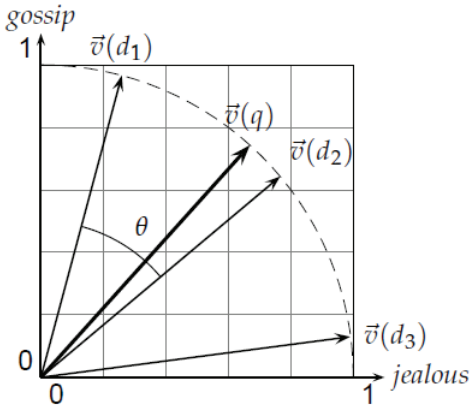



Figura 2.3: *Cosine similarity*: representación de los vectores. Fuente: [34]

Por lo tanto, esta semejanza se calcularía con la siguiente fórmula:

$$sim(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{\|\vec{q}\| \left\|\vec{d}\right\|} \tag{2.2}$$

donde el numerador representaría el producto escalar entre ambos vectores y el denominador es el producto de sus normas.

**Modelos probabilísticos** En 1977, Robertson presenta en [46] el Principio de Ranking Probabilístico (o *Probabilistic Ranking Principle*, en inglés) que sirvió de base para definir una familia de modelos a partir de este. En él se introduce una nueva forma de generar un ranking de los documentos en base

[3] Term Frequency - Inverse Document Frequency

a su probabilidad de ser relevante, asumiendo que la relevancia depende únicamente de la *query* y del documento. Además, se asume que los documentos estarían divididos en dos conjuntos: un conjunto $R$ con los documentos relevantes y un conjunto $NR$ de documentos no relevantes.

Según este principio, el caso más sencillo para crear un ranking de documentos sería devolver todos los documentos cuya probabilidad de ser relevante es mayor que la probabilidad de no serlo, es decir

$$d \text{ es relevante si } P(R = 1|d, q) > P(R = 0|d, q)$$

A pesar de lo sencillo que parece, se requiere conocer todas las probabilidades, algo que no pasa en un caso práctico. Por lo tanto, el Principio de Ranking Probabilístico no presenta ninguna forma concreta de calcular las probabilidades ni de crear un ranking de los documentos. A partir de este principio, surgen diferentes modelos que tratan de calcular estas probabilidades y crear un ranking de los documentos. Una gran desventaja de estos modelos es la necesidad de recalcular las probabilidades cada vez que un nuevo documento se inserta a la colección, haciendo que no sea el enfoque más eficiente.

El modelo más usado tradicionalmente ha sido el Modelo de Independencia Binaria (o *Binary Independence Model*, BIM, en inglés) [34]. Para ser capaz de calcular las probabilidades, este modelo asume

- que los documentos y las *queries* están representadas en forma binaria, como en el Modelo Booleano.
- que los términos son independientes entre sí. Al igual que en el Modelo del Espacio Vectorial, esta asunción es errónea ya que está claro que hay palabras dependientes entre sí y la aparición de algunas harán más probable la aparición de otras.
- que la relevancia de un documento es independiente del resto de relevancias de los documentos.
- que únicamente afectan a la probabilidad los términos que se encuentran en la *query*.

Otro modelo ha sido BM25, un esquema de ponderación de términos que permite eliminar la suposición de representación binaria de los documentos y las *queries*. En este modelo es posible usar las métricas usadas en el VSM, como el tf-idf, para ponderar los términos. Este modelo es más usado en problemas reales, ya que BIM había sido diseñado para búsquedas en un contexto controlado y pequeño.

TAN o *Tree Augmented Naive-Bayes* también se ha usado como modelo probabilístico con el fin de eliminar la suposición de que los términos son independientes entre sí. En este modelo, se pueden modelizar las dependencias entre términos en forma de árbol.

#### 2.2.4.2. Modelos modernos

A partir de los tres modelos tradicionales, surgieron nuevos modelos que buscaban mejorar de alguna manera su versión tradicional. Un ejemplo de estas variaciones es el Modelo del Espacio Vectorial Generalizado o GVMS. Este modelo surgió para corregir la suposición de su versión tradicional de que los términos de los documentos y las *queries* son independientes entre sí. Sin embargo, usar este modelo puede suponer un incremento en el tiempo de búsqueda [4].

Los Modelos de Lenguaje o *Language Models* son usados en numerosas tareas del PLN como la etiquetación morfológica o la traducción automática. Si en los modelos probabilísticos se calcula la probabilidad de relevancia de un documento dada una *query* ($P(R = 1|D, q)$) en los Modelos de Lenguaje

se trata de modelizar probabilísticamente el contenido de los documentos [34]. Una vez modelizados los documentos, estos son ordenados en orden de relevancia según la probabilidad de generarse la *query* a partir de ellos, o lo que es lo mismo: $P(q|M_d)$ siendo $M_d$ el modelo del documento $d$.

Las redes neuronales también se han usado como modelo para la RI destacando cuatro posibles usos [38]:

1. Estimar la relevancia de un documento y una *query* dadas unas representaciones manuales de estos (Véase Figura 2.4a).
2. Estimar la relevancia de un documento y una *query* a partir de unas representaciones de ambos extraídas de manera automática, como en la Figura 2.4b.
3. Generar las representaciones de los documentos y de las *queries* para después hallar su similitud con, por ejemplo, la similitud del coseno (Figura 2.4c).
4. Finalmente, para expandir una *query* antes de aplicar técnicas tradicionales de RI (Véase Figura 2.4d).

También se han usado otras técnicas como modelo para la RI como pueden ser las redes bayesianas [1] o técnicas de modelado de temas como *Latent Semantic Indexing* [19] y *Latent Dirichlet Allocation* [59].

## 2.3. Técnicas de expansión

Las técnicas de expansión de términos nos permiten añadir y completar la *query* con palabras relacionadas con el fin de aumentar la precisión y la cobertura a la hora de realizar la búsqueda documental. Como hemos visto en la Sección 2.2, estos métodos se usan tanto en la pre como post-traducción en la Búsqueda Documental Monolingüe y Multilingüe.

En concreto se va a hablar de tres métodos empleados para expandir los términos: *Relevance Feedback, Pseudo-relevance Feedback* o *Local Feedback* y *Local Context Analysis*.

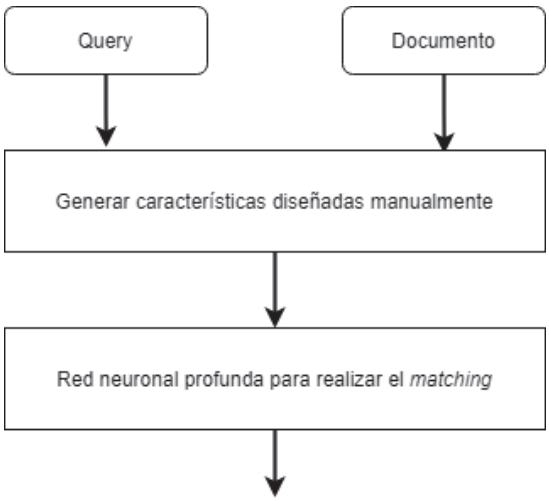
### 2.3.1. *Relevance Feedback*

*Relevance Feedback* [51] es un método en el que la *query* es modificada usando información extraída de los documentos más relevantes devueltos. Para ello, el usuario deberá seleccionar de los documentos devueltos cuáles considera relevantes y la *query* original se modifica con los $k$ términos más relevantes de los documentos marcados como relevantes por el usuario.

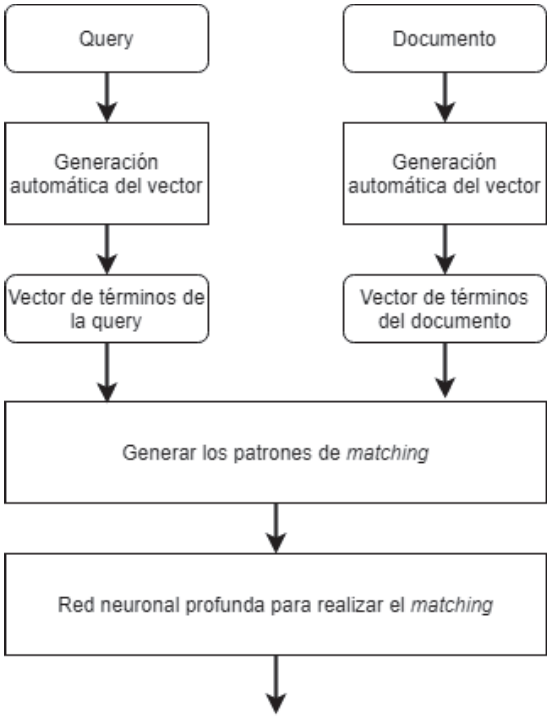
Imaginemos que estamos en la situación del ejemplo de la Figura 2.5. La *query* original tiene pesos de (0.7, 0.3) en relación a los ejes *retrieval* e *information* respectivamente. En el supuesto caso que el usuario marcase el $D_1$ como relevante, la *query* se vería modificada a (0.2, 0.8) por los pesos de este documento y se acercaría más al $D_1$. Por el contrario, si el usuario marcase el $D_2$ como documento relevante, se modificaría a (0.9, 0.1) y se acercaría al $D_2$. En este caso, al tratarse de un ejemplo que utiliza el VSM, esta acción de acercamiento se traduciría en una similitud del coseno mayor y por tanto mayor relevancia.

Como podemos ver, este método requiere de una intervención extra del usuario a la hora de la búsqueda para seleccionar los documentos que él considera relevantes:

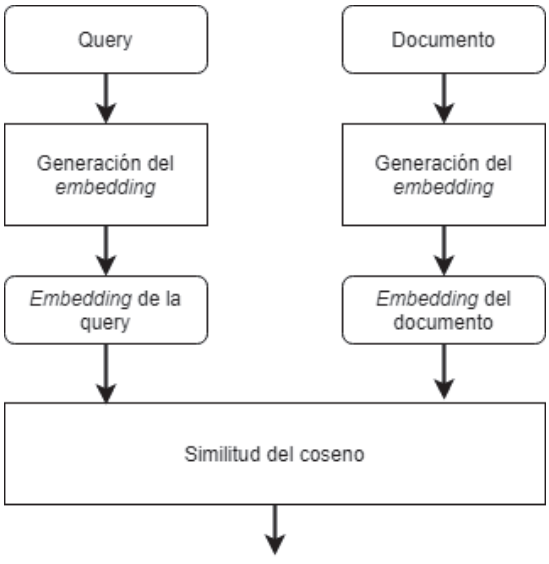
1. El usuario realiza una consulta y el sistema le devuelve una serie de documentos que considera relevantes.



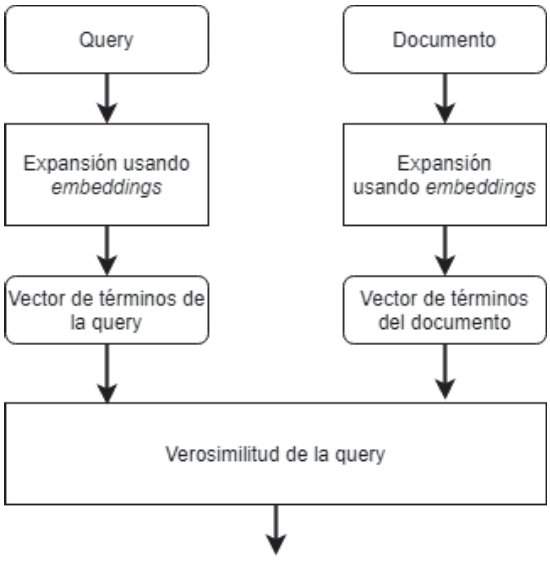

(a) Estimación de la relevancia a partir de representaciones manuales.




(b) Estimación de la relevancia a partir de representaciones automáticas.




(c) Generación de las representaciones.




(d) Expansión de la *query*.

Figura 2.4: Enfoques de modelos neuronales en la RI. Fuente: [38].

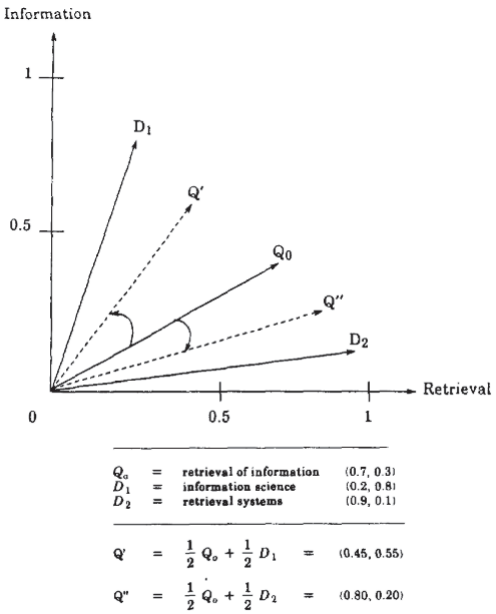



Figura 2.5: Ejemplo de *Relevance Feedback*. Fuente: [51].

2. El usuario marca cuáles de ellos considera relevante y se modifica la *query* en base a ellos, p. ej., añadiendo a la *query* original los términos más representativos de los documentos marcados.
3. finalmente el sistema le devuelve otro conjunto de documentos revisado en base a la similitud con la *query* modificada.

Lo mismo ocurriría en un sistema multilingüe (Véase Figura 2.6), con la principal diferencia que el usuario realizaría la consulta en una lengua O y el sistema CLIR le devolvería una serie de documentos en la lengua D. Este método puede no ser el más óptimo ya que obliga al usuario a marcar la relevancia o no de un documento en una lengua que no es la suya y puede que ni se sienta cómodo ni tenga la suficiente capacidad para juzgar la relevancia de un documento.

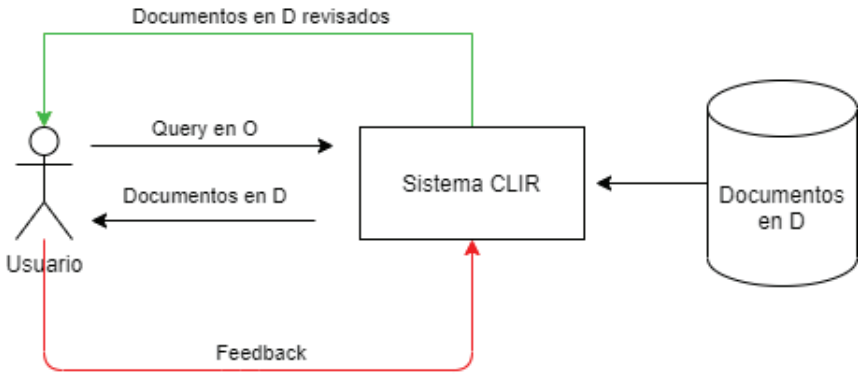



Figura 2.6: Ciclo de *Relevance Feddback* en un sistema CLIR.

### 2.3.2. *Local Feedback*

*Local Feedback* o *Pseudo-relevance Feedback* [3] funciona exactamente igual que el método anterior pero ahorra al usuario ese paso de marcar los documentos que considera relevantes al considerar los *k* primeros documentos devueltos como relevantes. De esta forma, utiliza para la expansión los

términos más importantes de estos documentos escogidos de manera automática.

### 2.3.3. *Local Context Analysis*

*Local Context Analysis* o Análisis Local del Contexto es un método de expansión de la *query* que no solo hace uso de feedback local sino también de información global, combinando ambas informaciones en el análisis [61].

Lo que se consigue con este método es beneficiar a los términos que co-ocurren frecuentemente con los términos de la *query* a la vez que se penalizan los términos que ocurren frecuentemente en la colección de documentos y se enfatizan los términos infrecuentes de la *query*.

Ballesteros y Croft demostraron en [6] que se obtenían mejores resultados al usar LCA frente a *Local Feedback* en la post-traducción. Sin embargo, en la pre-traducción se obtenían mejores resultados con *Local Feedback*. Pero la mayor ventaja de este método es que consigue niveles altos de precisión en niveles bajos de cobertura, es decir, los primeros documentos devueltos son relevantes. Este hecho es muy importante en los sistemas CLIR ya que el usuario puede no tener un gran nivel de lectura en la lengua de los documentos, siendo importante minimizar el número de documentos que el usuario tiene que comprobar.

## 2.4. Técnicas de traducción

El objetivo de este trabajo es crear un buscador documental multilingüe que nos permita buscar desde una lengua documentos en otra. Para ello, el enfoque más utilizado en la literatura ha sido el de introducir un módulo de traducción antes de realizar la búsqueda permitiendo superar así la barrera del lenguaje.

Además, es de vital importancia que este módulo funcione bien, pues de la calidad de la traducción dependerán los resultados de la búsqueda. Es por esto que la principal línea de investigación en sistemas CLIR haya sido la traducción.

Siguiendo la clasificación propuesta por Oard en [41], se pueden diferenciar dos grandes grupos de técnicas de traducción: basadas en el conocimiento o basadas en corpus.

### 2.4.1. Traducción basada en el conocimiento

Las técnicas de traducción basadas en el conocimiento afrontan la tarea de traducir haciendo uso de alguna fuente externa de conocimiento, p. ej. diccionarios o tesauros. La principal ventaja es que no dependen de la existencia de grandes colecciones multilingües de texto, únicamente de recursos externos que, por lo general, se encuentran disponibles en la mayoría de idiomas.

#### 2.4.1.1. Diccionarios

La idea básica de usar diccionarios para la traducción es la de traducir cada término de la *query* en la lengua de origen por un término apropiado o conjunto de términos en la lengua destino. A pesar de lo sencillo que es aplicar esta técnica, se obtienen resultados bastante buenos cuando se usa en conjunto con otra serie de técnicas, que veremos más adelante, para precisamente paliar sus desventajas.

**Enfoque básico** Los primeros en empezar a experimentar con esta técnica fueron Hull y Grefenstette [25] y Ballesteros y Croft [5]. Los primeros realizaron una serie de experimentos en un sistema

CLIR donde los documentos estaban en inglés y las *queries* se hacían en francés. Los segundos usaron documentos en inglés y realizaban consultas en español.

Hull y Grefenstette seleccionaban como traducción todos los posibles significados de una traducción. Por otra parte, Ballesteros y Croft únicamente seleccionaban como traducción el primer significado de la palabra traducida. Ninguno de los dos usó en la versión más simple ninguna técnica de expansión de términos.

Si nos fijamos en el ejemplo de la Tabla 2.3, la traducción de la *query* "*troubles civils*" sería "*turmoil discord trouble unrest disturbance disorder civil civilian courteous*" para los primeros y "*turmoil civil*" para los segundos.

Tabla 2.3: Ejemplo de traducción palabra por palabra usando diccionario. Fuente: [25].

| | |
|---|---|
| **Inglés:** | civil disturbances |
| **Francés:** | troubles civils |
| **Traducción:** | |
| | trouble - turmoil discord trouble unrest disturbance disorder |
| | civil - civil civilian courteous |

Con este enfoque básico, los primeros consiguieron un rendimiento del 59 % respecto al mismo sistema en su versión monolingüe y los segundos del 45 %. Ambos se dieron cuenta que sus enfoques no eran los mejores por una serie de motivos:

- Usar todas las posibles traducciones de una palabra introduce demasiada ambigüedad en la traducción, afectando a la precisión del sistema.
- Únicamente usar el primer sentido asumiendo que es el más importante es erróneo y puede ser el caso de que el término importante se quede sin traducir.
- Al traducir palabra por palabra se pierden expresiones de más de una palabra, p. ej. "*pommes de terre*", que significa patatas pero si se traduce palabra por palabra sería "manzanas de tierra". Está claro que
- Hay casos en los que una palabra no aparece en el diccionario, teniendo que incluir en la *query* final esa palabra sin traducir.

Para comprobar estas hipótesis, ambos realizaron unos experimentos en los que traducían a mano cada palabra, es decir, de todas las posibles traducciones escogían ellos la mejor; y otros en los que además seleccionaban las expresiones que aparecían y las traducían a mano. Comprobaron que con ambos enfoques mejoraba el rendimiento destacando un rendimiento del 90 % y del 86-98 % al traducir manualmente las expresiones.

**Expansión de la *query*** Viendo que los resultados obtenidos al usar diccionarios para traducir no eran buenos, también se aplicaron técnicas de expansión de términos tanto antes como después de traducir la *query* para intentar mejorar el rendimiento.

- En [5] se demostró que el uso de *Local Feedback* mejoraba el rendimiento del sistema un 50 %, presentando como ventaja respecto a *Relevance Feedback* que no requiere de una intervención extra por parte del usuario.

  Además, demostraron que usar esta técnica antes de la traducción tendía a mejorar la precisión mientras que usarla después de la traducción mejoraba la cobertura. Esto sugería que aplicar feedback antes de la traducción creaba una mejor base para la traducción y aplicarlo

después de la traducción reducía en parte el efecto negativo causado por la ambigüedad de las traducciones.

- También se comparó la efectividad de *LCA* frente a *Local Feedback* en [6]. Se comprobó que el primero funcionaba mejor tanto antes como después de la traducción, dando lugar a precisiones más altas en niveles bajos de cobertura, característica muy importante en los sistemas CLIR. Usando LCA se conseguía mejorar el rendimiento un 65 %, frente al 50 % al usar *Local Feedback*.
- Casi 10 años después, Levow, Oard y Lesnik obtuvieron peores resultados al usar *Local Feedback* en la pre-traducción en un sistema CLIR inglés-chino [30]. También aplicaron una expansión de términos en la post-traducción, obteniendo mejores resultados tal como se había demostrado en [5], [6].

Por lo tanto, usar técnicas de expansión de términos no es una tarea trivial en el sentido de que esté asegurado una mejora del rendimiento, dependerá de las lenguas y será necesario un estudio previo específico para cada caso.

**Detección de expresiones** Tanto Hull y Grefenstette como Ballesteros y Croft se dieron cuenta de que usar solamente diccionarios no era un buen enfoque para traducir debido a que no eran capaces de detectar las expresiones. Ambos comprobaron manualmente que detectarlas era fundamental para conseguir aumentar el rendimiento de los sistemas CLIR basados en diccionarios.

Ballesteros y Croft incluyeron un diccionario específico de expresiones para intentar ser capaces de detectar estas expresiones [6]. Algunos de los ejemplos de usar este diccionario específico se pueden ver en la Tabla 2.4.

Tabla 2.4: Expresiones traducidas usando un diccionario específico.

| **Expresión original** | **Traducción** |
|---|---|
| *united nations* | Naciones Unidas<br>Organización de las Naciones Unidas |
| *member country* | los países miembros<br>los países afiliados<br>los países participantes<br>los países pertenecientes |

Sin embargo, comprobaron que no siempre se mejoraba el rendimiento del sistema al traducir las expresiones. Llegaron a la conclusión que conseguir traducir expresiones correctamente mejoraba notablemente el rendimiento pero, traducir expresiones incorrectamente causaba un efecto opuesto. Por lo tanto, la calidad de la traducción en las expresiones sería más importante que en los términos sueltos.

**Desambiguación de las traducciones** Uno de los principales problemas de los primeros enfoques usando diccionarios en la traducción era la ambigüedad introducida al no saber qué sentido de la traducción era el correcto. Además, también se dieron cuenta de la importancia de desambiguar no solo las traducciones de términos sueltos sino también las traducciones de expresiones.

- El enfoque más simple de todos fue el realizar una etiquetación morfológica de las palabras antes de traducir, proceso conocido como *Part-of-Speech Tagging* en inglés. De esta forma, las traducciones de una palabra deberían respetar la etiqueta morfológica. Ballesteros y Croft en

[7] aplicaron esta técnica para desambiguar las traducciones de términos simples y consiguieron una mejora del rendimiento del 22 %. Un ejemplo de desambiguación usando esta técnica se puede ver en la Tabla 2.5, donde se traduce correctamente la palabra en español "gatos" como "*cat*" en inglés debido a su POS.

Tabla 2.5: Desambiguación usando POS.

| **Palabra original** | **Raiz** | **Posibles traducciones** | **Traducción** |
|---|---|---|---|
| gatos (sust) | gat | cat (sus), crawl (verb) | cat (sus) |

- La forma de estructurar la *query* afecta de manera directa al rendimiento de la búsqueda y, si se hace de una forma concreta, se puede reducir la ambigüedad de las traducciones [42]. En concreto, agrupando todas las posibles traducciones de una misma palabra bajo un operador "sinónimos".

  De esta forma, a la hora de realizar la búsqueda, este operador trata las apariciones de los términos dentro de él como apariciones de un único pseudo-término cuya frecuencia de documento es la suma de las frecuencias de documento de cada palabra del operador. Esto reduce la importancia de las palabras infrecuentes provocando un efecto de desambiguación. Al juntar este método con el anterior, en [7] consiguieron mejorar el rendimiento en un 89 % al traducir términos simples respecto a no usar ninguna técnica de desambiguación.

- Para desambiguar las traducciones de expresiones, Ballesteros y Croft presentaron en [7] un método de desambiaguación basado en co-ocurrencias en el que solo se necesitaba un corpus en la lengua de destino. El no necesitar un corpus paralelo es lo que hizo a este método innovador frente a otros enfoques en los que sí se necesitaba usar un corpus paralelo para desambiguar [12], [13].

  La hipótesis en la que se basaron para crear este método fue que las traducciones correctas de los términos de la *query* deberían co-ocurrir en los documentos de la lengua de destino y que las traducciones incorrectas deberían tender a no co-ocurrir.

  Dados dos términos, obtienen todas las traducciones de ambos que preserven su POS. Después, generan todos los conjuntos $a, b$ tal que $a$ es una definición del primer término y $b$ del segundo. Para calcular la co-ocurrencia de cada conjunto se utiliza la métrica *em*:

$$em(a,b) = max(\frac{n_{ab} - En(a,b)}{n_a + n_b}, 0) \tag{2.3}$$

  donde $n_a$ y $n_b$ son el número de ocurrencias de $a$ y $b$ en el corpus y $n_{ab}$ es el número de veces que $a$ y $b$ aparecen juntos en una ventana de $t$ palabras. $En(a,b) = \frac{n_a n_b}{N}$ siendo $N$ el número de ventanas en el corpus.

  De esta forma, se calcula esta métrica para cada conjunto $a, b$ y el conjunto con el mayor valor es escogido como la traducción correcta. Haciendo uso de esta técnica en la pre-traducción junto a LCA en la post-traducción se consiguió alcanzar un rendimiento del 94 % respecto a la versión monolingüe.

- En el año 2000, Adriani presentó una extensión del método basado en co-ocurrencias de Ballesteros y Croft capaz de desambiguar cualquier término de la *query* [2]. Para ello, seleccionaban para cada término la traducción que tuviese más sentido con el resto de las posibles traducciones de los otros términos de la *query*. De esta forma seleccionaban traducciones de

tal manera que los sentidos de las traducciones estuviesen relacionados – o fuesen estadísticamente similares– entre ellos. Esto es un proceso muy costoso computacionalmente hablando debido a las múltiples iteraciones anidadas.

Para calcular la similitud entre dos posibles sentidos utilizaron el *coeficiente de similitud de Dice* cuya fórmula es:

$$SIM_{xy} = 2\frac{\sum_{i=1}^{n}(w'_{xi} \cdot w'_{yi})}{\sum_{i=1}^{n} w_{xi}^2 + \sum_{i=1}^{n} w_{yi}^2} \tag{2.4}$$

siendo $w_{xi}$ el peso del término $x$ en el documento $i$, $w_{yi}$ el peso del término $y$ en el documento $i$, $w'_{xi}$ es igual que $w_{xi}$ si el documento $i$ también contiene el término $y$, si no 0, $w'_{yi}$ es igual que $w_{yi}$ si el documento $i$ también contiene el término $x$, si no 0 y n es el número de documentos de la colección. El peso de un término en un documento era calculado usando la métrica *tf-idf*.

En sus experimentos usando el inglés y el indonesio consiguieron mejorar el rendimiento del sistema CLIR aplicando esta técnica, sacando más beneficio en las *queries* que contenían expresiones. Además, volvieron a comprobar la efectividad de expandir la *query* pero siempre y cuando se haya resuelto antes la ambigüedad.

- Gao y otros [21] propusieron una mejora de los métodos de desambiguación basados en co-ocurrencias de Adriani y Ballesteros y Croft, un modelo de co-ocurrencias decrecientes. Los anteriores enfoques trataban por igual todas las palabras dentro de una ventana; sin embargo, Gao y otros descubrieron que las palabras que se encuentran más cerca están más relacionadas entre sí y, por tanto, deberían ser más similares entre sí.

  Para calcular la similitud entre dos términos hacen uso de la información mutua entre los términos:

  $$MI(x,y) = P(x,y) \cdot log(\frac{P(x,y)}{P(x) \cdot P(y)}) \tag{2.5}$$

  donde

  $$P(x,y) = \frac{C(x,y)}{\sum_{x',y'} C(x',y')}, \text{ y } P(x) = \frac{C(x)}{\sum_{x'} C(x')}.$$

  $C(x,y)$ es el número de co-ocurrencias de los términos $x$ e $y$ en una ventana predefinida en la colección y $C(x)$ es la frecuencia de documento del término $x$.

  En sus experimentos usando el inglés y el chino obtuvieron una mejora al usar esta técnica llegando a alcanzar un rendimiento del 88 % respecto la versión monolingüe del sistema, siendo el mejor resultado hasta la fecha para un sistema CLIR inglés-chino.

**Diccionario específicos** Otra de las desventajas de usar diccionarios como instrumento de traducción en los sistemas CLIR era que si un término no aparecía en el diccionario se tenía que introducir en la *query* final sin traducir. Obviamente, este hecho disminuye notablemente el rendimiento de los sistemas.

Para el caso de sistemas CLIR en un dominio concreto, usando diccionarios específicos además de diccionarios genéricos se consigue reducir el número de palabras que quedan sin detectar y traducir [42].

Pirkola realizó experimentos buscando documentos en inglés desde finés y en finés desde inglés en un dominio médico. Se dieron cuenta que la mayoría de las palabras que quedaban sin traducir eran demasiado específicas para ser encontradas en un diccionario genérico, p. ej. la traducción al inglés de la palabra "*osteoporoosi*" no se encontraba en un diccionario genérico pero sí en uno específico del dominio médico.

Probaron tres posibles combinaciones a la hora de traducir con los diccionarios:

1. Usar únicamente el diccionario genérico para realizar la traducción.
2. Primero traducir con el diccionario específico y las palabras no detectadas son traducidas con el diccionario genérico.
3. Traducir con ambos diccionario a la vez, eliminando traducciones duplicadas.

Los mejores resultados se obtuvieron con la tercera combinación tanto para traducir de inglés a finés como de finés a inglés. Finalmente, acabaron concluyendo que el uso de un diccionario específico es positivo para los sistemas CLIR al reducir el número de palabras no traducidas, pero que la forma en la que se usan afecta al resultado.

#### 2.4.1.2. Tesauros

Otra posible fuente de conocimiento que se puede usar para la traducción son los tesauros. Los tesauros son un conjunto de palabras ordenadas y estructuradas de una forma concreta con relaciones entre ellas.

En el caso de los sistemas CLIR, estaríamos interesados en tesauros multilingües, capaces de relacionar términos en una lengua con términos en otra. Estos tesauros multilingües son de gran utilidad en tareas de búsqueda documental no solo por permitir traducir términos de una lengua a otra sino también por establecer relaciones entre términos pudiendo así expandir las *queries* en base a estas. De acuerdo con la norma UNE-ISO 25964-1 **iso** que regula la estructura y las relaciones de los tesauros multilingües, las principales etiquetas son las siguientes:

- **USE** o use: el término que sigue a la etiqueta debería ser usado preferentemente en lugar del término preferente.

  *automóviles* **USE** coches

- **UP** o usado por: el término que procede a la etiqueta no debería ser usado y en su lugar se debería usar el término que precede a la etiqueta.

  coches **UP** *automóviles*

- **TG** o término genérico: el término que procede a la etiqueta tiene un significado más amplio que el término que la precede.

  clase universitaria **TG** clase

- **TE** o término específico: el término que procede a la etiqueta tiene un significado más específico.

  literatura **TE** literatura clásica

- **TR** o término relacionado: el término que procede a la etiqueta está relacionado con el término que la precede pero sin ser sinónimos.

  inteligencia artificial **TR** aprendizaje automático

Sin embargo, el principal problema de estos tesauros multilingües es su limitada existencia y disponibilidad, siendo la mayoría muy específicos en un dominio, no sirviendo para tareas de búsqueda documental genérica. Además, la mayoría de estos tesauros no son realmente multilingües y se centran únicamente en unas pocas lenguas, siendo prácticamente bilingües. Un ejemplo muy interesante de un tesauro realmente multilingue es el proyecto HEREIN [23], en el cual se construyó un tesauro relacionando más de 500 conceptos sobre el patrimonio europeo en 15 lenguas distintas.

Otro punto a tener en cuenta es el cómo crear los tesauros: manualmente o de manera automática. Yang, Wood y Cutkosky [62] hicieron una serie de experimentos en los que compararon el rendimiento de un sistema de búsqueda documental monolingüe usando tesauros manuales y tesauros automáticos. Como era de esperar, llegaron a la conclusión de que se obtenían mejores resultados al usar tesauros manuales; sin embargo, los tesauros automáticos eran más fáciles de crear y mantener.

**Creación manual** Los primeros experimentos con un sistema CLIR fueron realizados por Salton a principio de los 70 [47], [49]. En ellos, utilizaba un tesauro multilingüe construido a mano para realizar una búsqueda documental en alemán de documentos en inglés y francés de documentos en inglés, respectivamente.

En ambos experimentos obtenía resultados esperanzadores, alcanzando prácticamente el mismo rendimiento que en las versiones monolingüe. A pesar de los buenos resultados, este escenario no era realista puesto que estaba trabajando con un vocabulario muy pequeño y controlado lo que posible la construcción manual de los tesauros, cosa que en un escenario real con un vocabulario muy grande no era factible.

Más de 20 años después de los experimentos iniciales de Salton, Soergel presentó una guía con una serie de pautas sobre cómo crear un tesauro multilingüe enfocado a sistemas CLIR [56]. De nuevo, teniendo como limitación principal que este enfoque solo es factible en sistemas con un vocabulario controlado que permitan la creación de tesauros manuales.

Es por ello que prácticamente se abandonase este enfoque y las principales líneas de investigación sobre el uso de tesauros multilingües para traducir en sistemas CLIR se centraran en la creación automática de estos recursos y cómo aplicarlos a sistemas CLIR.

**Creación automática** La creación manual de tesauros multilingües en escenarios reales donde el vocabulario no es controlado es una tarea infactible. Es por ello de la necesidad de técnicas que permitan crear tesauros multilingües de manera automática.

- Sheridan y Ballerini utilizaron una colección de documentos comparables de la agencia suiza de noticias en alemán e italiano para crear automáticamente un *tesauro de similitud* [54].

  Un *tesauro de similitud* es una estructura que permite reflejar cómo de similares son los términos en el dominio de la colección de documentos. Para conseguir esto, se invierte el esquema tradicional en el que los documentos son representados en base a sus términos, representándose en este caso los términos en función de los documentos en los que aparecen. Si buscamos un término de la *query* en este *tesauro de similitud* seríamos capaces de obtener una lista de términos similares (o relevantes) a este.

  Para tratar la colección de documentos, realizaban primero un alineado de las noticias basándose en su fecha y en sus *keywords*. De esta forma, en el archivo 240895.mil para el italiano estarían todas las noticias en italiano del día 24 de agosto de 1995 que tuviesen como etiqueta "*militar*". De igual forma, su homólogo alemán.

  Una vez creados estos archivos auxiliares, se juntaban los archivos con misma fecha y misma *keyword* en un único archivo. Como resultado, en el archivo 240895.mil estarían tanto las noticias en italiano como en alemán del 24 de agosto de 1995 que tratasen sobre "*militar*".

  A la hora de realizar la búsqueda en alemán de documentos en italiano, expandían la *query* original con términos obtenidos del *tesauro de similitud*. Filtraban las palabras quedándose únicamente con las palabras en italiano y realizaban la búsqueda de los documentos. De tal manera, esta expansión de términos producía un efecto de traducción.

- Un año más tarde, Sheridan, Braschler y Schäuble utilizaron un *tesauro de similitud* para realizar búsquedas documentales en el dominio legal [55]. Comprobaron que el corpus a partir del cual se crea el *tesauro de similitud* afecta a la calidad de las traducciones. Al crear el *tesauro de similitud* a partir del mismo corpus que sobre el que posteriormente se iban a realizar búsquedas se obtenían mejores resultados que si se utilizaba un corpus diferente al de las búsquedas para crear el tesauro.
- Otro posible enfoque fue el usar las co-ocurrencias entre los términos para crear automáticamente tesauros multilingües. Schütze y Pedersen utilizaron un *tesauro de co-ocurrencias* con el que se asociaba a cada término un vector de co-ocurrencias y se podía calcular la similitud entre estos vectores para sacar términos semánticamente similares [52] entre sí. Sin embargo, este enfoque solamente fue aplicado a un sistema monolingüe y no llegaron a probar su aplicación en un sistema de búsqueda documental.
- Un año más tarde, Brown creó un tesauro multilingüe basado en co-ocurrencias a partir de un colección de textos paralelos en inglés y español [8]. Además, propuso una mejora respecto al método anterior al introducir un paso de filtrado posterior a la extracción de las co-ocurrencias.

  También realizaron experimentos para comprobar si la longitud del corpus usado para la creación afectaba de manera directa a la calidad del tesauro generado y, por tanto, al rendimiento del sistema CLIR. Los resultados obtenidos fueron un poco desconcertantes al comprobar que se obtenían mejores resultados al crear el tesauro con un corpus más pequeño. Esta comprobación era totalmente opuesta al otro paradigma usando fuentes del conocimiento, donde estaba demostrado que al usar diccionarios de mayor tamaño y con más palabras el rendimiento de un sistema CLIR mejoraba.

**Uso de herramientas existentes** Si en el apartado anterior vimos enfoques cuyo principal objetivo era la creación automática de un tesauro multilingüe capaz de realizar la traducción de una lengua a otra, también hubo enfoques que buscaban aplicar directamente tesauros multilingües a sistemas CLIR.

- Eichman, Ruiz y Srinivasan utilizaron el tesauro multilingüe UMLS (*Unified Medical Language System* o Sistema Unificado de Lenguaje Médico) para realizar unos experimentos de búsqueda en español y en francés de documentos médicos en inglés [16].

  A la hora de traducir las *queries* al inglés, utilizan este tesauro ya creado como un diccionario con el que realizar la traducción. Por lo tanto, este método es bastante similar a los basados en diccionarios.

  Proponen tres estrategias distintas para traducir:

  1. *Full match*: únicamente cogen las traducciones de las entradas del tesauro que coinciden exactamente con las palabras de la *query*.
  2. *Partial match*: para cada palabra que queda sin traducir ordenan las entradas en el tesauro y se escogen aquella de menor longitud (después de eliminar las *stopwords*). Si varias entradas empatan, se escoge la que tengo menor identificador.
  3. *Word based translation*: para cada palabra de la *query* que queda por traducir, identifican las entradas en el tesauro que contienen esa palabra y extraen las traducciones. Después listan las palabras pertenecientes a las traducciones y escogen la más frecuente como la traducción.

4. Si no es posible traducir alguna palabra introducirla en la *query* final sin traducir.

También presentaron un método para calcular la similitud entre una palabra de la *query* y una entrada del tesauro usando el Coeficiente de Dice.

En estos experimentos lograron obtener un rendimiento de entre el 71 % - 79 % y 51 % - 61 % respecto a la versión monolingüe del sistema para español e inglés, respectivamente.

- Otro enfoque más moderno es el de Franco-Salvador, Rosso y Navigli en el que usaron BableNet, una red semántica multilingüe de términos, como fuente de conocimiento para realizar la traducción [18].

  Además, para calcular la similitud realizan dos procesos:

  1. Representan los documentos como grafos en los que los nodos son los términos que aparecen en el documento y las relaciones son inferidas de BableNet. Calculan una similitud parcial en base a los grafos ($S_g$) utilizando el Coeficiente de Dice.
  2. Representan los documentos como vectores y calculan la similitud parcial ($S_v$) usando la similitud del coseno.

  Una vez calculadas las dos similitudes parciales, las agregan en una única similitud tal que

$$KBSIM(d, d') = c(G) \times S_g(G, G') + (1 - c(G)) \times S_v(v, v') \tag{2.6}$$

  donde $c(G)$ es la densidad del grafo. De esta forma ponderan ambas similitudes de manera dinámica en función de la calidad del grafo.

  Este método fue usado para buscar documentos pero no en la forma que estamos acostumbrados a ver en los sistemas CLIR de buscar mediante una *query* documentos similares. Se realizó una búsqueda de documentos dando como entrada un documento en español y buscando los documentos más similares en inglés. A pesar de no ser exactamente igual, la adaptación de este método a un sistema CLIR sería una tarea trivial.

**WordNet** WordNet es una base de datos léxica de palabras en inglés [17]. En ella se agrupan sustantivos, adjetivos, adverbios y verbos según su significado creando grupos de sinónimos (o *synsets*, en inglés).

Debido a la agrupación de las palabras en grupos de significado parecido, parece una herramienta interesante para usar en tareas de búsqueda documental. Varios fueron los intentos de usar WordNet en tareas relacionadas con la RI; sin embargo, los resultados obtenidos fueron muy negativos debido a varios motivos, de los que destacamos [45], [58]:

1. WordNet no hace similitudes entre palabras con diferente POS. Debido a la propia estructuración de las palabras, estas están separadas según su etiqueta morfológica. Por lo tanto, es imposible calcular similitudes entre, por ejemplo, ”saltar” y ”salto”.
2. Las relaciones que aparecen en WordNet son del tipo hiperonimia, hiponimia o término hermano. Sin embargo, la mayoría de relaciones entre términos que podría establecer una persona no aparecen.
3. Directamente algunos términos no están incluidos en la base de datos.

**EuroWordNet** El homólogo multilingüe de Wordnet es EuroWordNet. Es una base de datos multilingüe creada a partir de *wordnets* en varios idiomas como el español o el italiano. De esta forma, es posible utilizar esta base de datos como fuente de conocimiento en sistemas CLIR.

Gonzalo, Verdejo y Chugir propusieron un sistema CLIR en el que se usaba EuroWordNet a la hora de traducir [22]. Hacían uso de los índices interlingüísticos que tiene EuroWordNet para indexar los documentos y así poder realizar *queries* desde cualquier lengua incluida en esta base de datos. Los resultados que obtuvieron fueron muy prometedores al obtener en la versión multilingüe el mismo rendimiento que la versión monolingüe sin indexar por conceptos. Sin embargo, ese enfoque se abandonó y no se propusieron.

### 2.4.2. Traducción basada en corpus

El otro gran grupo de técnicas de traducción en sistemas CLIR son aquellas que hacen uso de la información de un corpus multilingüe para llevar a cabo la traducción. Se pueden encontrar dos tipos:

1. **Corpus paralelos**: en este tipo de corpus se tiene un conjunto de documentos en varios idiomas exactamente iguales. Este tipo de corpus es el más buscado por los investigadores debido a que evita tener que alinear los documentos; sin embargo, también es el más escaso debido a la complejidad de crear un corpus de este tipo. Un ejemplo sería la Wikipedia, donde podemos encontrar las mismas páginas en varios idiomas.

2. **Corpus de documentos comparables**: para los casos en los que no es posible usar un corpus paralelo, se utiliza un corpus de documentos comparables. Este corpus está formado por documentos similares, no teniendo por qué ser exactamente iguales, bastando que estén relacionados de alguna forma. Por ejemplo, una colección de noticias internacionales de un periódico español y uno inglés se podría considerar un corpus comparable.

A raíz de estos dos tipos de corpus multilingües, surgen dos principales enfoques en la literatura sobre sistemas CLIR traduciendo con información del corpus: el primero, sería cómo traducir teniendo información de un corpus multilingüe; y el segundo, sería como alinear un corpus en el caso de tener documentos comparables.

**Usos de corpus paralelos** Dunning y Davis proponen usar el VSM con un corpus paralelo para realizar la traducción [15]. De esta forma, creaban una especie de matriz de traducción con la que eran capaces de traducir las *queries*. A pesar de funcionar bastante bien, era demasiado complejo computacionalmente por lo que descartaron este enfoque.

También se usó programación evolutiva a partir de información de un corpus paralelo para refinar y mejorar las traducciones de las *queries* [36]. Sin embargo, no se llegó a aplicar este método en un sistema CLIR sino que más bien se evaluaba la calidad de las traducciones.

Sabiendo de las limitaciones de usar diccionarios como método de traducción en sistemas CLIR, surgieron enfoques en los que se usaba información de corpus paralelos para desambiguar las traducciones producidas por los diccionarios. Estos enfoques consiguen reducir en gran parte la ambigüedad introducida por los diccionarios [12].

No obstante, no siempre se obtienen buenos resultados al desambiguar con información de corpus paralelos. En esos casos, etiquetar morfológicamente las palabras antes de la desambiguación consigue solucionar estos problemas [13].

**Alineación de corpus comparables** Cuando no se dispone de un corpus paralelo y se tiene un corpus de documentos comparables, es importante alinear estos documentos antes de utilizarlos tanto para traducir como desambiguar.

La mayoría de corpus comparables suelen tratar de noticias internacionales en varias lenguas no teniendo por qué ser exactamente iguales. El enfoque más sencillo es aprovecharse de la naturaleza de estos corpus y hacer uso de descriptores que incorporan los textos para alinearlos. En el caso de las noticias, es común que contengan la fecha en la que se publicaron y una serie de etiquetas que describen su temática. A pesar de la simpleza de esta técnica, se obtienen buenos resultados [54].

Sin embargo,hay veces en los que este enfoque no funciona bien. Cuando ocurre esto, es necesario utilizar una técnica de alineado más compleja. En [57], extraen de cada noticia sueca los términos más representativos, los traducen y lanzan estas *queries* contra las noticias en inglés. Para realizar el alineado se basan en la fecha y en la similitud obtenida consiguiendo alinear un 18 % de los documentos de la colección.

Otro enfoque distinto para alinear los documentos, es usar los temas de los documentos en vez de características internas como su longitud, fecha... Usando técnicas probabilísticas de modelado de temas (*Topic Modelling*, en inglés) se obtiene un mejor rendimiento en el alineado [64].

Por regla general, usar únicamente la información extraída de corpus multilingües no es suficiente para obtener una calidad óptima en los sistemas CLIR y es necesario complementar estas técnicas con otras como los diccionarios. Además, a pesar de obtenerse buenos resultados cuando se combinan varias técnicas, la disponibilidad de estos corpus multilingües es escasa en comparación con, por ejemplo, los diccionarios, teniendo que a veces hacer uso de colecciones comparables lo que introduce una complejidad extra al sistema CLIR al tener que alinear los documentos.

### 2.4.3. Otras técnicas de traducción

En este apartado veremos otra serie de técnicas de traducción que no están incluidas en ninguna de las dos grandes categorías explicadas anteriormente, pero no por ello, dejan de ser importantes. Con todas ellas se han hecho ya experimentos usándolas para traducir en sistemas CLIR salvo con UNL *(Universal Networking Language)* con el que solamente se han presentado formas de usarlo como método de traducción.

#### 2.4.3.1. Traducción automática

Utilizar traducción automática para traducir las *queries* parece ser la forma más lógica de superar la barrera del lenguaje en un sistema CLIR, tal como podemos ver en la Figura 2.7.

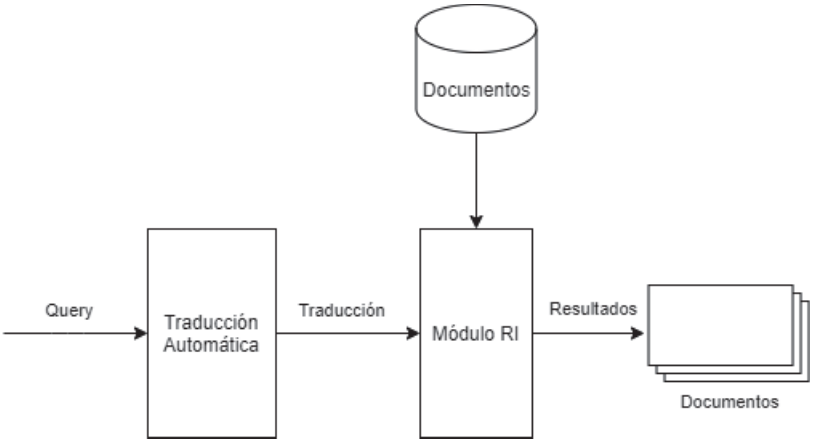



Figura 2.7: Sistema CLIR con traducción automática.

Sin embargo, este enfoque es muy criticado por los investigadores [27]. Las principales razones que

destacamos son:

- La calidad de la mayoría de los sistemas de traducción automática no es del todo buena, acentuándose la baja calidad de las traducciones con textos cortos en los que falta contexto, justo como son las *queries*.
- En relación con lo anterior, los sistemas de traducción automática intentan que las traducciones sean sintácticamente correctas. Este hecho de nuevo no es aplicable a sistemas CLIR donde las *queries* suelen ser gramaticalmente incorrectas.
- Por último, implementar un sistema de traducción automática es caro en comparación con otros enfoques como son el uso de diccionarios o tesauros.

#### 2.4.3.2. Interlinguas

Otro posible enfoque en los Buscadores Documentales Multilingües sería el uso de una interlingua como representación intermedia entre varias lenguas. Las principales características que definen una interlingua son las siguientes [9]:

1. Se busca encontrar una representación del significado de las palabras común a varias lenguas.
2. Una interlingua no deja de ser otra lengua en el sentido de que es autónoma y necesita por tanto unos componentes definidos: vocabulario y relaciones semánticas.
3. En una interlingua las unidades semánticas atómicas son los sentidos y no las palabras.
4. Las relaciones temáticas y funcionales se establecen entre las unidades semánticas atómicas.

Los enfoques presentados hasta el momento buscan traducir bien sean los documentos o las *queries* de una lengua a otra directamente. Esta forma de traducir provoca una explosión de combinaciones de traducciones si realmente se quiere ser fiel al término multilingüe. Por el contrario, el uso de una interlingüa sería más práctico y eficiente ya que se reduciría el número de traducciones necesarias en un ambiente multilingüe. En la tabla 2.6 se puede ver un ejemplo del número de pares de lenguas que habría en el caso de no usar una interlingua (primera columna) o usando una interlingua (segunda columna). Si no se usase una interlingua, la progresión de crecimiento de los pares de lengua sería de $n(n-1)$ mientras que si se usase una interlingua sería solamente de $2n$.

Tabla 2.6: Crecimiento del número de pares de lenguas en función del uso o no de una interlingua.

| Número de lenguas | Pares de lenguas no interlingua | Pares de lengua interlingua |
|---|---|---|
| 2 | 2 | 4 |
| 3 | 6 | 6 |
| 4 | 12 | 8 |
| 5 | 20 | 10 |
| 6 | 30 | 12 |
| 7 | 42 | 14 |
| 8 | 56 | 16 |
| 9 | 72 | 18 |
| 10 | 90 | 20 |

Al usar una interlingua, simplemente se tendrían que realizar dos traducciones para cada lengua:

1. Traducir de la lengua X a la interlingua.

2. Traducir de la interlingua a la lengua X.

Esta diferencia en el número de pares de lenguas entre usar o no usar una interlingua sería aún mayor a medida que aumentásemos el número de lenguas del sistema, haciendo de la interlingua un enfoque mucho más práctico en ambientes multilingües.

**UNL: *Universal Networking Language*** En [10], [11] se presenta la interlingua *Universal Networking Language* como solución a la multilingualidad en dos aplicaciones concretas como son la extracción de la información para crear un sistema multilingüe de pregunta-respuesta y la creación de ontologías multilingües.

Esta interlingua fue creada a finales de los años 90 por la Universidad de las Naciones Unidas con el fin de crear una representación genérica del lenguaje independiente de las lenguas y así conseguir superar las barreras del lenguaje. UNL está formado principalmente por tres componentes:

1. **Universal Words (UWs)**: forman el vocabulario de la interlingua UNL. Estas palabras universales están formadas por una palabra en inglés (*headword*) y por una serie de restricciones que definen un sentido único para cada palabra, reduciendo así al mínimo la ambigüedad.

   Se puede hablar de tres tipos de restricciones en base a sus funciones:

   - **Restricciones ontológicas**: sirven para clasificar las UWs y definen una estructura jerárquica. Por ejemplo, la palabra universal *bus(icl>vehicle)* haría referencia a que *vehicle* es un hipernombre de *bus*.
   - **Restricciones semánticas**: sirven para restringir el sentido de las palabras. En el caso de la palabra inglesa ”bank”, se pueden hablar de dos posibles significados: orilla o banco económico. Para desambiguar el significado sería tan simple como incluir una restricción de la siguiente manera:
     - ”bank” de orilla: *bank(icl>side)*
     - ”bank” de banco económico; *bank(icl>financial_institution)*
   - **Restricciones argumentales**: sirven para indicar los argumentos que puede tener un verbo en una oración. Por ejemplo, el verbo comer puede tener como argumentos quién como y qué come, por lo que sería *eat(icl>do, agt>person, obj>food)*.

2. **Relaciones**: sirven para establecer las relaciones entre los conceptos de UNL, es decir, las palabras universales. De entre todas, se destacan las siguientes:

   - **Relaciones causales**: sirven para expresar la razón, propósito u objetivo o condición de las acciones.
   - **Relaciones argumentales**: al igual que las restricciones argumentales, indican las relaciones de un verbo con el resto de partes de la oración.
   - **Relaciones circunstanciales**: sirven para expresar las circunstancias de una acción, es decir, la manera en la que se realiza un evento, el instrumento con el que se ejecuta una acción, etc.
   - **Relaciones nominales**: sirven para expresar modificaciones a los nombres como puede ser el poseedor de una cosa.
   - **Relaciones temporales**: sirven para definir el tiempo en el que se lleva a cabo una acción o el tiempo en el que un evento comienza.

- **Relaciones espaciales**: se usan para definir los lugares donde ocurren los eventos, bien sean reales o virtuales.

3. **Atributos**: sirven para expresar la información contextual y van precedidos por el caracter "@". Ejemplos de atributos serían *@entry* para connotar el nodo de entrada en un grafo UNL o *@pl* para expresar número plural.

#### 2.4.3.3. Aprendizaje profundo

Las técnicas de aprendizaje profundo son aquellas que usan algún tipo de red neuronal para resolver un problema. En la actualidad, representan el estado del arte en muchas aplicaciones de la Visión por Computador o el Procesamiento del Lenguaje Natural tales como el reconocimiento de objetos, segmentación de imágenes o generación del lenguaje.

En el año 2013 se presentó *Word2Vec*, un nuevo modelo neuronal capaz de aprender las representaciones de las palabras de un corpus [37]. Este modelo supuso una revolución en el campo del PLN y sirvió de entrada para otras muchas aplicaciones, como la RI. Juntando este hecho con el auge de técnicas de aprendizaje profundo, en los últimos años la mayor parte de la literatura sobre sistemas CLIR se centra en aplicar estas técnicas para traducir las *queries* de una lengua a otra y realizar la búsqueda.

Uno de los inconvenientes de usar técnicas de aprendizaje profundo es que se necesitan conjuntos de datos multilingües muy grandes para entrenar los modelos y sean capaces de obtener buenos resultados. Además, en el caso de tareas multilingües como lo son los sistemas CLIR, al igual que con las técnicas de traducción basadas en corpus es necesario que los datos en ambas lenguas estén alineados de alguna forma, aumentando aún más la complejidad del problema.

Para solucionar este problema, un enfoque muy interesante es realizar una alineación de manera no supervisada [31]. La principal ventaja de este enfoque frente a otros es que evita tener que usar colecciones bilingües en los que los datos estén alineados bien sea nivel de palabra o a nivel de documento. El método que emplean es muy similar al propuesto en 1993 por Davis y Dunning [15]: aprenden una matriz de traducción que sirve para mapear los *word embeddings* en la lengua origen con los de la lengua destino.

Otro de los principales problemas es que no se ha llegado a implementar aún ningún sistema CLIR completo con estas técnicas. En [26] usan BERT *(Bidirectional Encoder Representations from Transformers)*, un modelo de lenguaje basado en técnicas de aprendizaje profundo que usa la arquitectura *Transformer* y analiza las frases en ambos sentidos, para realizar unos experimentos simulando un sistema CLIR en lituano e inglés. Hacemos uso del término "simulación" debido a que sus experimentos simplemente se limitaban a calcular la similitud entre un conjunto de palabras (la *query*) y una frase (simulando un documento), no llegando a implementar una búsqueda documental como tal.

Además, estas técnicas dependen completamente de un modelo que ha sido entrenado para un tarea muy concreta y para unas lenguas determinadas. Si se le suma el hecho de que entrenar estos modelos es computacionalmente muy costoso, provoca que las técnicas de aprendizaje automático no sean la mejor opción en cuanto a escalabilidad si se quisiese construir un sistema verdaderamente multilingüe.

## 2.5. Motores de búsqueda actuales

Si bien es cierto que nuestro trabajo se trata de crear un buscador documental multilingüe ”offline”, entendiendo ”ofline” como que se realiza la búsqueda sobre un conjunto cerrado de documentos almacenados localmente y no sobre un conjunto abierto y en la Web, no por ello se tienen que obviar a los principales motores de búsqueda de Internet. Como se explicó al comienzo de este Estado del Arte para diferenciar entre buscador documental y motor de búsqueda, los motores de búsqueda son aquellos buscadores que realizan sus búsquedas en la Web a través de Internet.

A día de hoy, los cuatro motores de búsqueda más usados en el mundo son [53]:

1. **Google**: con un porcentaje de uso del 92.41 %, estando muy por encima del resto de competidores.
2. **Bing**: con un porcentaje del 2.46 %. Este es un buscador desarrollado por Microsoft y hacen uso de la búsqueda semántica para mejorar sus resultados [33].
3. **Yahoo**: con un 1.48 % de uso.
4. **Baidu**: con un 1.3 %. Este es un motor de búsqueda en chino muy popular en ese país.

A pesar de ser los referentes en el ámbito de los motores de búsqueda, si se comprueban cada uno de ellos no ofrecen la posibilidad directa de buscar páginas webs en un idioma desde otro. Sí que tanto Google como Bing al realizar una búsqueda en una lengua extranjera, intentan hacer un mix de resultados entre páginas en la lengua de la búsqueda y páginas en la lengua de tu ubicación. Sin embargo, la calidad de estos resultados está muy lejos de ser óptima.

## 2.6. Conclusiones

La Búsqueda Documental Multilingüe es un área donde en los últimos años también se han tratado de aplicar técnicas de aprendizaje profundo debido a su demostrada superioridad en otras áreas frente a las técnicas clásicas. Sin embargo, por el momento no se ha llegado a una solución real aplicando estas técnicas y se han limitado a realizar experimentos a pequeña escala. Además, presentan una serie de desventajas como son el coste de entrenamiento de los modelos, la falta de generalidad al depender exclusivamente de un modelo y la necesidad de colecciones gigantescas de datos para poder entrenar los modelos.

Otras técnicas como la traducción automática no tienen mucha aceptación entre la comunidad debido a la baja calidad que se obtiene al traducir frases pequeñas formadas por palabras sueltas y sin una estructura gramatical predefinida, como son las *queries*. El uso de información extraída de corpus paralelos no es una buena opción a no ser que se complemente con otro método como por ejemplo los diccionarios. Además, estas técnicas, al igual que las de aprendizaje profundo, requieren de grandes colecciones multilingües de texto.

El uso de diccionarios plantea un enfoque sencillo, transparente y disponible para la gran mayoría de lenguas. No obstante, usar diccionarios requiere del uso de algún método externo como complemento, capaz de desambiguar las traducciones para conseguir buenos resultados. Los tesauros suponen una aproximación válida al poder aprovechar la forma en la que están estructuradas y relacionadas las palabras entre sí para hallar similitudes entre ellas. No obstante, no resulta fácil encontrar tesauros válidos para la búsqueda documental lo que obliga en muchos casos a crear tesauros de manera automática obteniendo peores resultados que con tesauros manuales o ya existentes.

# Capítulo 3

# Planteamiento del problema

Como se ha visto hasta ahora, a día de hoy no se ha llegado a una solución definitiva en la Búsqueda Documental Multilingüe. Ninguna técnica parece haberse consolidado en esta difícil tarea, pues no solo se ha de realizar una búsqueda documental, sino también superar las barreras del lenguaje para permitir buscar documentos en una lengua desde otras.

Además de esto, existe otra gran cuestión y es que inicialmente se planteó como un problema multilingüe que permitiese la búsqueda de documentos desde varias lenguas. Sin embargo, la gran mayoría de técnicas CLIR expuestas en el capítulo anterior acaban convirtiendo el problema original en un problema bilingüe; limitándose, en el mejor de los casos, a realizar una búsqueda entre 3 o 4 lenguas, muy lejos del concepto multilingüe.

Es por ello que se plantea el uso de la interlingua *Universal Networking Language* como posible solución al problema de la multilingualidad en los buscadores documentales. Mediante el uso de esta técnica, se obtendrían dos beneficios respecto al resto de técnicas planteadas hasta el momento:

1. Supondría una traducción desambiguada debido a la propia naturaleza de UNL.
2. Se podría realizar realmente un buscador documental multilingüe al no depender de la existencia de otros recursos en cada lengua ni de un modelo entrenado para una tarea concreta.

El uso de UNL para resolver el problema de la multilingualidad en aplicaciones del Procesamiento del Lenguaje Natural no es algo nuevo, ya que se ha planteado como solución para sistemas de pregunta-respuesta y para crear ontologías multilingües. A pesar de esto, sí supondría un enfoque novedoso en los sistemas CLIR.

Por lo tanto, se hará uso de UNL para realizar una representación intermedia de las palabras en los documentos y en las *queries* para, posteriormente, aplicar un modelo de búsqueda documental sobre las palabras universales creadas.

# Capítulo 4

# Hipótesis de trabajo

En este capítulo se mencionará un conjunto de restricciones que hemos establecido a la hora de crear nuestro buscador documental multilingüe:

- Para realizar la traducción se utilizará la interlingua *Universal Networking Language*.
- No se almacenarán los documentos representados en UNL sino que solamente se guardarán en español.
- Solamente se podrán realizar *queries* en español o en inglés para buscar los documentos en español.
- El sistema no incluirá ningún mecanismo de auto detección de la lengua de la *query*. Por lo tanto, el usuario deberá indicar la lengua en la que realiza la consulta en el momento de la búsqueda.
- No se usará ningún método de expansión de la *query* y solamente se usará como método sencilla de desambiguación la etiqueta morfológica de las palabras.
- Por razones de tiempo, no se creará un diccionario concreto con palabras universales a partir de los documentos del sistema sino que se utilizará un diccionario ya creado con la falta de calidad que ello puede suponer.
- Para los experimentos se utilizará una colección pequeña formada por 75 *queries* y 200 documentos.
- No hay número máximo de términos en la *query*, pero se limitará el tamaño de las *queries* a 1 o 2 términos.
- La validación de los experimentos se realizará de manera manual, es decir, no se contará con tablas de relevancia sino que se irá viendo *query* a *query* si los documentos mostrados son o no relevantes.

# Capítulo 5

# Propuesta de modelo

En esta sección nos centraremos a nivel conceptual en las diferentes funcionalidades que ha de incluir nuestro modelo para poder resolver el problema planteado en esta tesis, poder buscar documentos en una lengua desde varias lenguas.

Al tratarse de un buscador documental, lo primero que se deberá hacer será realizar un cálculo de la relevancia entre la *query* introducida por el usuario y el conjunto de documentos almacenados. Por relevancia se entiende cómo de importante es un documento dada una *query*, por ejemplo, si la *query* es *partidos de fútbol* los documentos que hablen sobre partidos de fútbol serán muy relevantes.

Además, al ser un buscador documental multilingüe se tendrá que, de alguna manera, realizar una traducción para poder realizar búsquedas de documentos en una lengua desde otras.

El modelo que proponemos se ajusta al siguiente diagrama (véase Figura 5.1), donde podemos observar tres partes diferenciadas en el modelo: un preprocesamiento, un conversor multilingüe de la *query* y finalmente un buscador.

## 5.1. Preprocesamiento del texto

Para poder realizar la conversión multilingüe y la posterior búsqueda, se tiene que preprocesar el texto de un manera en la que pueda ser tratado posteriormente.

Para ello, se realizarán una serie de procesos con el fin de limpiar y transformar el texto. En concreto, se plantea el uso de tokenización, para separar en palabras el texto original; eliminación de palabras vacías y normalización de las palabras, para conseguir una uniformidad en las palabras de cara a la traducción posterior; y, además, la eliminación de caracteres sin significado y el paso a minúsculas de las palabras.

**Tokenización** La tokenización es la tarea encargada de romper un texto original en partes más pequeñas, llamadas *tokens*. En nuestro caso particular, al aceptar nuestro sistema únicamente texto en inglés y español, la división de un texto en *tokens* se puede hacer de manera muy sencilla limitándonos a separar el texto por los espacios en blanco ya que son, de manera implícita, los separadores naturales de las palabras.

De esta manera, el texto original ”partidos de primera división” quedaría dividido en partes más pequeñas

[partidos, de, primera, división]

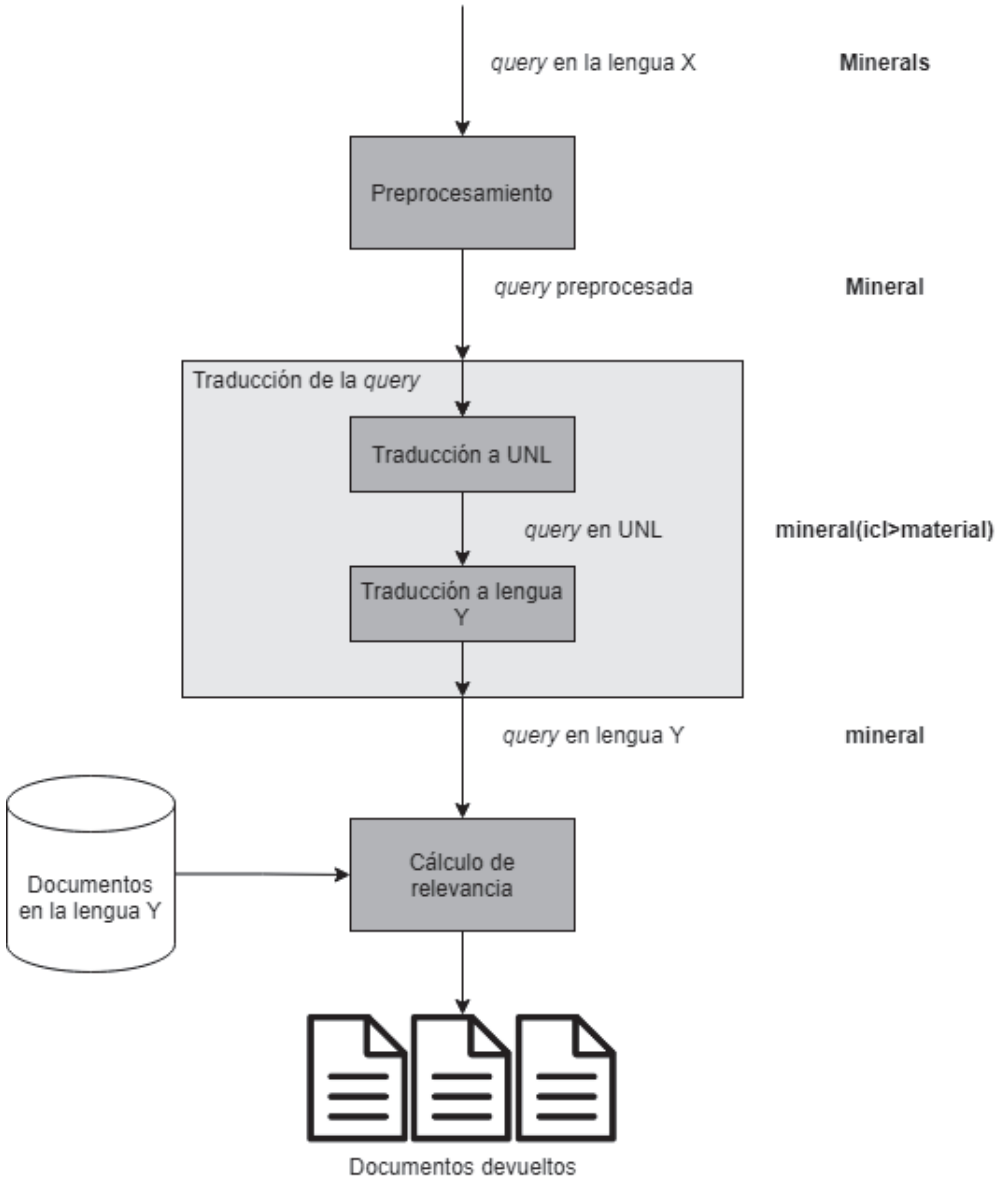



Figura 5.1: Flow completo de nuestro modelo.

**Eliminación de las palabras vacías** La siguiente tarea a realizar consistiría en eliminar aquellos tokens que no aportan significado al texto, como las preposiciones, verbos auxiliares, etc. Para ello se utiliza una lista de palabras vacías o *stopwords*, filtrando los *tokens* y eliminando aquellos *tokens* que aparecen en la lista creada.

Al eliminar las palabras vacías, el resultado del ejemplo anterior sería

[partidos, primera, división]

Como se plantea en la traducción, a la hora de preprocesar la *query* no se eliminarán las palabras vacías para aumentar así la precisión de la traducción, al poder contener expresiones del tipo "lugar de trabajo" que se traduce como "*workplace*" en vez de palabra por palabra "*place of work*".

**Normalización** El siguiente paso a realizar en nuestro preprocesamiento del texto sería normalizarlo. A la hora de buscar documentos, parece razonable que una búsqueda con el texto "partidos de primera división" sea igual a una búsqueda con el texto "partido de primera división". Esto se consigue por medio de la normalización.

A la hora de normalizar las palabras, las dos formas más usadas son el *stemming* y la lematización.

- El ***stemming*** consiste en aplicar una serie de reglas heurísticas encargadas de reducir las palabras a sus raíces. Hay que tener en cuenta que el uso de esta técnica no garantiza que la palabra resultado sea una palabra existente ni que mantenga la misma etiqueta morfológica que la original.

  A continuación se puede ver un ejemplo de la aplicación de esta técnica de normalización en donde las palabras resultantes sí existen pero no todas mantienen la misma categoría morfológica:

  $$\begin{aligned} \text{program (sustantivo)} &\Longrightarrow \text{program (sustantivo)} \\ \text{programs (sustantivo)} &\Longrightarrow \text{program (sustantivo)} \\ \text{programmer (sustantivo)} &\Longrightarrow \text{program (sustantivo)} \\ \text{programming (verbo)} &\Longrightarrow \text{program (sustantivo)} \\ \text{programmers (sustantivo)} &\Longrightarrow \text{program (sustantivo)} \end{aligned}$$

- La **lematización** es una técnica más compleja al no basarse en reglas heurísticas sino que utiliza el vocabulario propio de una lengua para reducir las palabras a sus lemas. Al contrario que con el *stemming*, la lematización sí garantiza que la palabra resultante exista y sea de la misma categoría morfológica que la original.

  Un ejemplo del resultado de la lematización sería el siguiente:

  $$\text{saltó (verbo)} \Longrightarrow \text{saltar (verbo) saltarines (sustantivo)} \Longrightarrow \text{saltarín (sustantivo)}$$

Tras probar ambos métodos, se decidió **usar la lematización** al preservar la etiqueta morfológica de las palabras y garantizar que la palabra resultante de la normalización existe, hecho de vital importancia como podremos ver más adelante a la hora de traducir las palabras.

Como resultado de esta normalización, el ejemplo propuesto en este apartado quedaría de la siguiente manera:

[partido, primero, división]

**Etiquetación morfológica o *Part of Speech Tagging*** A la hora de realizar la traducción, nos dimos cuenta de que sería interesante contar con la etiqueta morfológica de cada *token* para realizar una desambiguación muy básica en el que caso de que una palabra tenga varias posibles traducciones.

Tras analizar varias herramientas, se decidió usar el modelo *en_core_web_trf* para inglés y el *es_dep_news_trf* para español de spaCy [24]. Ambos modelos tienen una precisión del 98 % al realizar la etiquetación morfológica.

Como resultado de esta etiquetación, cada token iría acompañado de su etiqueta morfológica, pudiendo ser cualquiera de las siguientes:

- VERB: verbo.
- SYM: símbolo.
- SCONJ: conjunción subordinante.
- PUNCT: signo de puntuación.
- PROPN: sustantivo propio.
- PRON: pronombre.
- PART: partícula, sirve de comodín cuando no se consigue etiquetar una palabra con ninguna otra categoría.
- NUM: número.
- NOUN: sustantivo.
- INTJ: interjección.
- DET: determinante.
- CCONJ: conjunción coordinante.
- AUX: verbo auxiliar.
- ADV: adverbio.
- ADP: preposición.
- ADJ: adjetivo.
- X: otros.

Por lo tanto, el resultado final de preprocesar nuestro ejemplo ”partidos de primera división” sería

[ partido:NOUN, primero:ADJ, división:NOUN ]

## 5.2. Conversión multilingüe de la *query*

Una vez preprocesada la *query*, ya se puede empezar con la conversión multilingüe. Como se ha visto en el Estado del Arte, hay tres posibles enfoques en los buscadores documentales multilingües a la hora de traducir: traducir la *query* en el momento de realizar la búsqueda, traducir los documentos al indexar, o traducir ambos.

Se plantea como solución traducir la *query* debido a las ventajas explicadas en el Estado del Arte, entre las que destacamos:

- Rapidez a la hora de realizar la traducción al ser textos de unas pocas palabras frente a textos más grandes como son los documentos.
- No hay necesidad de multiplicar el espacio de almacenamiento traduciendo y almacenando los documentos en varias lenguas.

Asimismo, para conseguir superar la barrera del lenguaje en el buscador documental, se plantea el uso de una interlingua como solución debido a las ventajas que aportaba frente al resto de soluciones planteadas hasta el momento:

- Normalmente, solo vemos aplicaciones ”multilingües” (cuando en verdad son bilingües) para pares de lenguas comunes en las que se encuentra mucho material disponible, como el inglés, francés, español o chino. Sin embargo, en lenguas más pobres en cuanto a material disponible, esos enfoques no valdrían ya que dependen de grandes cantidades de materiales.

  Es precisamente en estos ambientes donde el enfoque de la interlingua cobra fuerza al representar de una manera desambiguada cada término, siendo más fácil así realizar la traducción

a esas lenguas. Por tanto, este enfoque garantiza una fiabilidad extra a la hora de traducir en lenguas exóticas.

- Tal como se explicó en la Sección 2.4.3.2, en el caso de aplicaciones verdaderamente multilingües el uso de una interlingua reduce de manera drástica el número de traducciones a realizar, necesitándose únicamente para cada nueva lengua 2 traducciones: lengua-interlingua e interlingua-lengua.

En concreto, se hará uso de la interlingua **Universal Networking Language** para realizar las traducciones debido a ser una interlingua con mucho renombre (creada por las Naciones Unidas), usada en otras tareas del PLN y estar ya demostrada su eficacia en otras aplicaciones [9]-[11].

Esta interlingüa representa las palabras por medio de **palabras universales, o Universal Words (UW)** en inglés, por ejemplo:

$$bank(icl > financial\_institution)$$

donde la primera palabra es la ***headword*** y el resto son una serie de **restricciones** semánticas que permiten identificar inequívocamente a cada palabra universal.

Por lo tanto, se plantea usar UNL como puente entre la lengua de la *query* y la lengua de los documentos. De esta forma, se necesitaría una traducción a UNL para cada lengua en la que fuese posible realizar las *queries* y otra traducción de UNL a la lengua de los documentos, en nuestro caso, el español. Un esquema de este proceso se puede observar en la Figura 5.2.

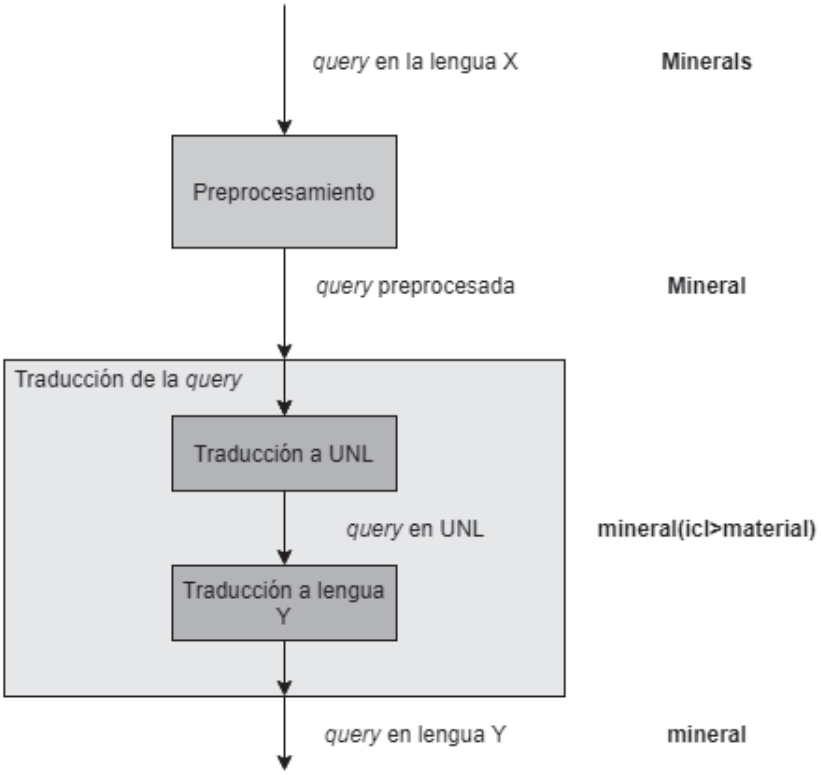



Figura 5.2: Conversión de la palabra de la *query* "*minerals*" en inglés al español mediante la interlingua UNL.

Estas traducciones se harían mediante el uso de diccionarios, concretamente de dos tipos de diccionarios:

1. **Diccionario lengua X a UNL**: para cada lengua en la que fuese posible realizar las *queries* crearemos un diccionario. Este diccionario sería de la forma $< término\ en\ lengua\ X >:< UW >$.

   Un ejemplo de posible entradas de este diccionario para el inglés serían las siguientes:

   $$mineral : mineral(icl > material)$$
   $$bank : bank(icl > financial\_institution)$$

2. **Diccionario de UNL a lengua Y**: en nuestro caso, se creará un diccionario de UNL a español, para así conseguir convertir términos desde cualquier lengua al español.

   $$mineral(icl > material) : mineral$$
   $$bank(icl > financial\_institution) : banco$$

No nos bastará con realizar traducciones de la *query* palabra por palabra sino que además se tendrá que intentar traducir grupos de palabras. Imaginemos que la *query* del usuario contiene el término en español "lugar de trabajo". En el ejemplo de la Tabla 5.1 podemos ver que si se realizase la traducción de cada palabra de manera individual se obtendría una traducción errónea del término que no representaría su significado.

Tabla 5.1: Ejemplo de traducción de palabras de manera individual vs de manera conjunta.

| **Palabras** | **Traducción de palabras individuales** | **Traducción de palabras por grupos** |
|---|---|---|
| lugar, de, trabajo | spot(icl>point>thing,equ>topographic_point)<br>task(icl>work>thing,equ>undertaking) | workplace(icl>geographic_point>thing) |

Para conseguir este funcionamiento se hace uso de un algoritmo basado en [60], el cual se puede describir a muy alto nivel de la siguiente manera:

1. Si la *query* en inglés tiene una sola palabra, se busca esa palabra en el diccionario, se traduce y se va al Paso 3. Si esa palabra no se encuentra en el diccionario quedaría sin traducir.

2. Si la *query* tiene varias palabras, se obtienen todas las combinaciones secuenciales posibles de la *query*.

   [lugar, de, trabajo] = [lugar, de, trabajo, lugar de, de trabajo, lugar de trabajo]

   *a*) De mayor a menor tamaño de combinaciones, se intenta buscar una entrada en el diccionario y se convierte.

   *b*) Si una combinación se consigue traducir, se eliminan de las combinaciones calculadas en el Paso 2 todas aquellas que contienen alguna palabra perteneciente a la combinación traducida.

   **Ejemplo** Siguiendo con el ejemplo, empezaríamos buscando una entrada para el término "lugar de trabajo". Suponiendo que este término no tiene una entrada en el diccionario, seguiríamos intentando traducir el resto de combinaciones. La siguiente sería "de trabajo", que en este caso supongamos que sí tiene una entrada en el diccionario, por lo que se traduce. A continuación, eliminaríamos las combinaciones restantes que tengan alguna palabra de la combinación que se acaba de traducir. Aplicando este proceso, el conjunto de combinaciones quedaría de la siguiente forma

   [lugar]

al haberse eliminado todas las combinaciones que contengan las palabras ”de” y ”trabajo”.

**Desambiguación** Al usar un diccionario, puede darse el caso de que un término tenga varias entradas en el diccionario y, por tanto, pueda tener varias traducciones posibles.

Debido a este hecho, es necesario introducir algún proceso de desambiguación que nos permita saber qué entrada corresponde al significado concreto del término en la *query*. Para ello, se incluye un método muy sencillo de desambiguación que consiste en sustituir únicamente las entradas cuya categoría morfológica coincide con con la categoría del término en la *query*. Este proceso es conocido como *POS Disambiguation*.

## 5.3. Buscador

Una vez que tenemos la *query* en la misma lengua que los documentos, ya podemos realizar la búsqueda mediante el cálculo de la relevancia entre la *query* del usuario y cada uno de los documentos, para finalmente mostrarle al usuario los $k$ más relevantes.

**Medida de similitud** A la hora de calcular la relevancia de cada documento con la *query* del usuario se hará uso de una medida de similitud.

Varias son las opciones de medidas de similitud, tales como la similitud de Jaccard o la distancia Euclídea entre dos vectores. Sin embargo, se plantea como posible solución el uso de la similitud del coseno (o *cosine similarity*) debido a su velocidad a la hora de realizar el cálculo y a su buen funcionamiento en tareas de recuperación de la información. La fórmula de la similitud del coseno es

$$sim(query, documento) = \frac{query \cdot documento}{|query| \cdot |documento|}$$

donde *query* es el vector que representa a la búsqueda del usuario y *documento* es el vector que representa a cada documento.

De esta forma, se obtendría un valor de relevancia para cada documento pudiendo así después ordenarlos de mayor a menor y mostrar al usuario los $k$ más relevantes.

**Extracción de características** Para convertir las *queries* y los documentos en un formato entendible a la hora de aplicar la similitud del coseno, proponemos la métrica tf-idf para extraer las características de los textos.

Usar esta métrica frente a, por ejemplo, únicamente la frecuencia, tiene como principal ventaja que no solo se tienen en cuenta las palabras más frecuentes en los documentos (las que aparecen más veces) sino también aquellas poco comunes (las que aparecen en pocos documentos). De esta forma, se consigue valorar cada palabra de una manera mucho más realista no basándose únicamente en la frecuencia bruta.

Imaginemos el caso de un conjunto de documentos sobre noticias de fútbol. La palabra ”gol” es muy probable que aparezca muchas veces pero también aparezca en la mayoría de noticias de la colección, por lo que no sería tan discriminante. Sin embargo, imaginemos que tenemos una noticia concreta sobre la gala del balón de oro. En esa noticia es muy probable que la palabra ”premio” aparezca muchas veces y aparezca en pocas noticias a parte de en esta. En este caso sí se puede decir que la palabra ”premio” es representativa y tiene suficiente poder discriminante en la noticia sobre la gala del balón de oro. Precisamente este efecto lo conseguimos al hacer uso de la métrica tf-idf.

Además, esta métrica es muy fácil de calcular al limitar las operaciones a multiplicaciones y una división y aporta una buena precisión en las búsquedas.

# Capítulo 6

# Experimentación y resultados

En este capítulo se explicará la experimentación realizada para validar el modelo propuesto en el capítulo anterior, empezando por su diseño previo y la implementación realizada para acabar con el análisis de los resultados obtenidos en las diversas pruebas realizadas.

## 6.1. Diseño de experimentación

**Obtención de documentos** Para validar nuestro modelo, en primer lugar se necesita un conjunto de documentos sobre el cual realizar las *queries*. Por lo tanto, el primer paso de nuestra experimentación ha sido recopilar un conjunto de 200 noticias de varios diarios españoles [20]. En concreto, estas noticias se han sacado de cuatro temáticas distintas: política, salud, deporte y tecnología.

En la Figura 6.1 se puede ver que la longitud de estas noticias va desde las 100 palabras hasta unas 600 las que más, siendo las más comunes aquellas con entre 150-250 palabras.

**Extracción terminológica** Una vez obtenido nuestro conjunto de documentos de prueba, se deberá crear un conjunto de *queries* con las que probar nuestro modelo. Para ello, primero se ha realizado una extracción terminológica de los 50 términos más relevantes de las noticias, mediante el uso de un extractor terminológico propio [20]. Estos términos pueden estar formados tanto por una única palabra como por varias.

Más en concreto, este extractor terminológico mezcla tres métodos distintos a la hora de realizar la extracción. Comentando muy por encima este extractor, ya que no es el cometido de este trabajo, podemos decir que hace lo siguiente:

1. Realiza una extracción terminológica usando el algoritmo de clustering *k-means*. Aprovechándonos que son noticias de 4 temáticas distintas, podemos calcular las palabras más representativas de cada cluster, realizando así una extracción parcial de cada categoría y agregándolas después.

2. También realiza una extracción terminológica usando la métrica tf-idf, consiguiendo así sacar las palabras más representativas del conjunto de noticias.

3. Hacemos uso también de un extractor terminológico ya desarrollado de la librería **gensim**[44].

4. Finalmente, agregamos las tres extracciones en una sola y realizamos un filtrado manual del conjunto final.

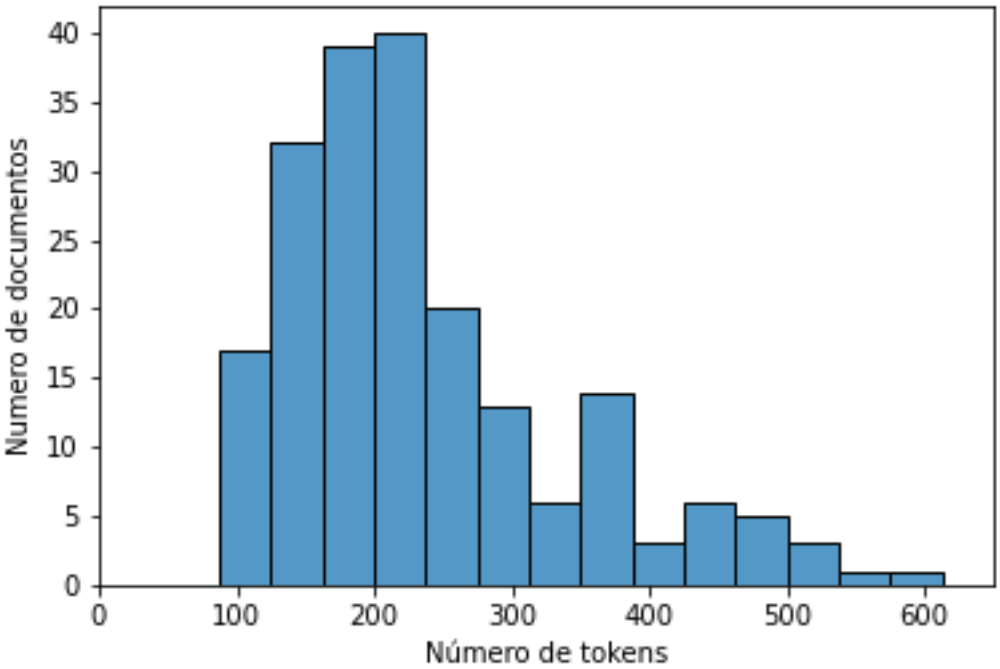



Figura 6.1: Análisis del número de *tokens* de las noticias.

Por lo tanto, mediante un proceso semiautomático se consiguen extraer los 50 términos más representativos de nuestro conjunto de documentos.

**Traducción de los términos extraídos** El objetivo de esta Tesis Fin de Máster es ser capaces de buscar documentos en una lengua desde otra y para ello se necesitarán *queries* en una lengua distinta que los documentos. Por lo tanto, lo que se ha hecho ha sido una traducción manual a inglés de los términos extraídos mediante el extractor terminológico.

**Creación del conjunto de *queries*** Una vez tenemos los términos extraídos tanto en inglés como en español, se puede crear un conjunto de *queries* con las que evaluar el rendimiento de nuestro sistema CLIR. En los buscadores documentales las *queries* suelen ser muy cortas, de 1-3 palabras [4]. Es por ello que para evaluar nuestro sistema se usarán *queries* de 1 y 2 términos.

Para ello, se emplearán como *queries* los 50 términos extraídos y se hará una mezcla de ellos para formar otro conjunto de 25 *queries* formadas por 2 términos.

**Métrica de bondad** Finalmente, una vez tenemos nuestro conjunto de documentos y nuestro conjunto de *queries* se puede validar el funcionamiento de nuestro sistema.

Esta validación consistirá en usar una determinada métrica de bondad. La precisión y la cobertura son métricas de bondad muy usadas en problemas de Aprendizaje Automático. Sin embargo, en el campo de la Recuperación de la Información surgen otro tipo de métricas como la $P@n$ (precisión media tras devolver $n$ documentos) o MAP (*Mean Average Precision* o promedio de la precisión media). Este tipo de métricas surgen de la necesidad de evaluar el rendimiento de un sistema en determinadas posiciones del ranking, especialmente las primeras.

Además, en los buscadores documentales no se suelen requerir valores altos de cobertura si no más bien valores altos de precisión, ya que interesa que los documentos de las primeras posiciones sean realmente relevantes [4]. Si le sumamos el hecho de ser un buscador documental multilingüe, esta precisión alta en las primeras posiciones es aún más buscada debido a las limitaciones lingüísticas por parte del usuario [41].

Por lo tanto, para evaluar nuestro modelo **haremos uso de la P@3**, es decir, la precisión en la posición 3 del ranking o lo que es lo mismo: el número de documentos relevantes entre los 3 primeros.

**Experimentos** Una vez definidas todas las estructuras y la métrica de bondad a usar, los experimentos que se llevarán a cabo serán los siguientes:

1. **Búsqueda monolingüe vs búsqueda multilingüe**: en los sistemas CLIR es común evaluar el rendimiento del sistema comparándolo con su versión en español y viendo cómo de diferente es el rendimiento en términos relativos. Por lo tanto, se ejecutará el conjunto de 100 *queries* en español (ESP2ESP) y se verá cómo mejora o empeora al buscar los documentos con *queries* en inglés (ENG2ESP).
2. **Calidad de la traducción**: también se hará un experimento para comprobar si realmente la calidad de la traducción usando UNL, y en especial el diccionario que se usa, es buena.

## 6.2. Implementación

Para llevar a cabo la experimentación propuesta en la sección anterior, se ha realizado la siguiente implementación[1] usando el lenguaje de programación Python. Además, se han usado varias librerías conocidas en el ámbito del PLN para agilizar el desarrollo del proyecto:

- **Gensim** [44]: esta librería la hemos usado a la hora de la indexar los documentos y de calcular la relevancia de la *query* con los documentos.
- **spaCy** [24]: esta librería la hemos utilizado a la hora de realizar el análisis morfológico de los *tokens*.
- **NLTK** [32]: esta librería la hemos usado en varías tareas del preprocesado como la tokenización.

La estructura de nuestro modelo, ya organizada en distintos módulos de implementación, se puede observar en la Figura 6.2. A su vez, podemos observar dos procesos distintos que nuestro sistema es capaz de hacer:

1. **Indexación de los documentos**: este proceso es el encargado de guardar una representación de los documentos entendible de cara a realizar la búsqueda después. Para ello primero deben ser preprocesados y finalmente almacenados.
2. **Búsqueda**: este proceso es el encargado de buscar los documentos relevantes para una *query* del usuario. Al igual que con los documentos, primero se ha de preprocesar la *query*, traducirla mediante UNL a español y realizar la búsqueda sobre la colección de documentos. Finalmente, al usuario se le mostrarían los $k$ documentos más relevantes en función de la *query* introducida.

A continuación, detallaremos qué acciones realiza cada módulo de la implementación, coincidiendo la mayoría con las distintas funcionalidades explicadas en la Sección 5 y daremos detalles específicos de la implementación realizada.

[1] https://github.com/themrcesi/tfm

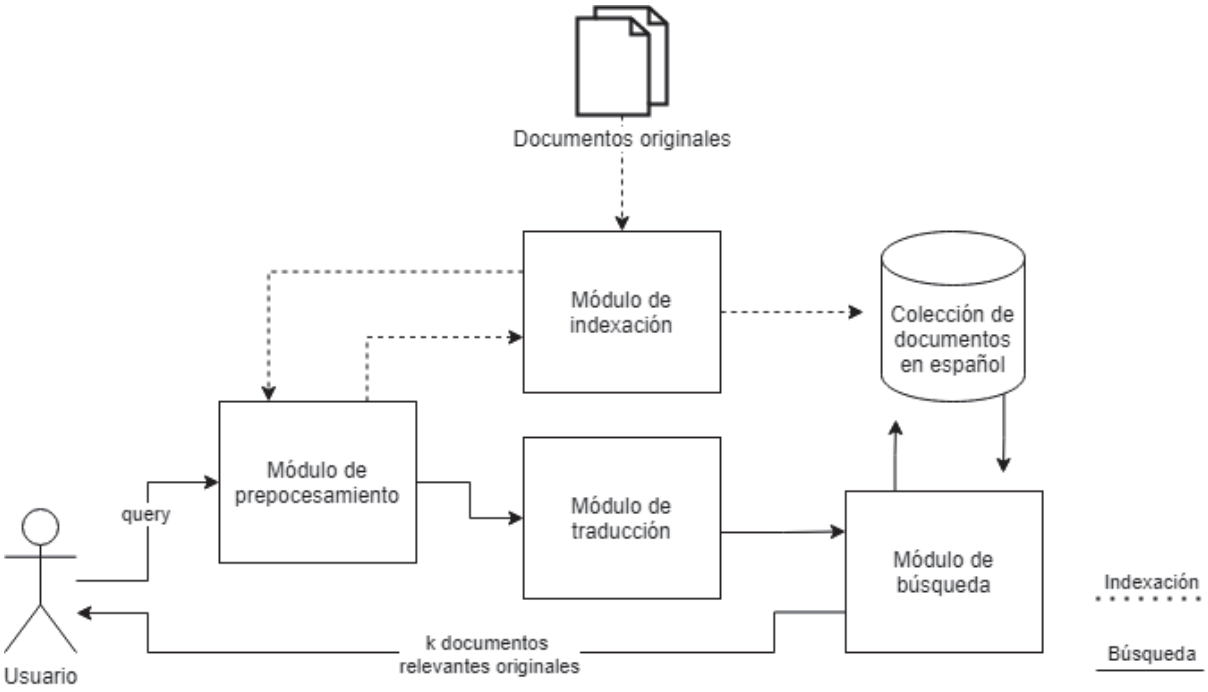



Figura 6.2: Diseño de nuestro buscador documental multilingüe.

### 6.2.1. Módulo de preprocesamiento

El módulo de preprocesamiento es el encargado de realizar todas las tareas explicadas de tratamiento del texto. En concreto, hemos dividido este módulo en 4 submódulos más pequeños e independientes entre sí (véase Figura 6.3) mediante el patrón de diseño "Decorador". De esta forma, cada submódulo es independiente al módulo previo y posterior, pudiendo intercambiarse entre sí de una manera rápida y sencilla.

Además, el usuario deberá indicar la lengua de la consulta ya que en función de la lengua se le aplicará un modelo de lematización y etiquetación morfológica diferente. Es importante destacar cómo a la hora de preprocesar la *query* no se eliminan las palabras vacías para mejorar así la precisión de la traducción posterior, tal como se explicó en la Sección 5.2 del modelo conceptual.

A continuación podemos ver el código necesario para crear las *pipelines* de preprocesamiento en español para los documentos (que incluye todos los pasos) y en inglés y español para las *queries* (sin la eliminación de las palabras vacías):

```
# pipelines queries preprocessing
pipeline_query_eng = LemmatizerTagger("es_dep_news_trf", Tokenizer(SymbolRemover()))
pipeline_query_esp = LemmatizerTagger("en_core_web_trf", Tokenizer(SymbolRemover()))

# pipeline documents preprocessing
pipeline_doc = LemmatizerTagger("es_dep_news_trf", StopwordsRemover(
    "../resources/spanish_stopwords.txt", Tokenizer(SymbolRemover())))
```

### 6.2.2. Módulo de indexación

El módulo de indexación es el encargado de almacenar los documentos de una manera en la que después puedan ser tratados de manera eficiente en el módulo de búsqueda a la hora de calcular las similitudes. Esta indexación se realiza por medio de los *archivos invertidos (o índices invertidos)*, que

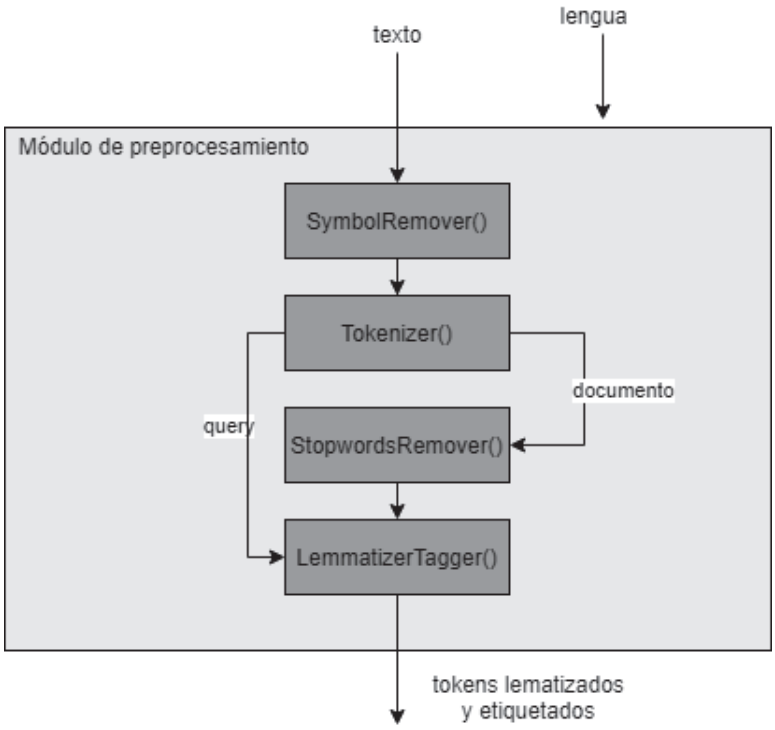



Figura 6.3: Estructura modo *pipeline* del módulo de preprocesamiento.

son un mecanismo para indexar texto con el fin de acelerar la tarea posterior de búsqueda [4].

Este mecanismo está formado por dos elementos:

- **El vocabulario o diccionario**: que es el conjunto de palabras de la colección.
- Las **ocurrencias** de las palabras.

La forma más sencilla de indexar una colección de documentos es mediante la matriz término-documento (*term-document* matrix, en inglés). Las filas de esta matriz están formadas por los términos de la colección, es decir, por el diccionario o el vocabulario; y las columnas están formadas por los documentos de la colección. Cada celda de esta matriz tiene como valor el número de veces que aparece el término i-ésimo en el j-ésimo documento.

En la Tabla 6.1 se puede ver un ejemplo reducido de esta matriz, en la que las filas representan el vocabulario de la colección (4 palabras) y las columnas los documentos de la colección (2 documentos). Por ejemplo, la celda de la primera fila y la primera columna significa que la palabra "comida" aparece 5 veces en el documento #1.

Tabla 6.1: Ejemplo de matriz término-documento.

| | **D1** | **D2** |
|---|---|---|
| **comida** | 5 | 2 |
| **hamburguesa** | 2 | 3 |
| **pizza** | 1 | 1 |
| **perrito caliente** | 0 | 1 |

A la hora de implementar este módulo, se ha hecho uso de las clases **Dictionary** y **MmCorpus** del

módulo *corpora* de la librería **gensim**. A continuación puede verse la clase **Indexer** en donde hemos encapsulado las funcionalidades de preprocesar e indexar los documentos. Además, cabe destacar que, para acelerar esta tarea, el proceso de preprocesamiento de los documentos se realiza de manera paralela mediante la clase **Parallel** de la librería **joblib**.

```
class MyCorpus:
    def __init__(self, docs, dictionary):
        self.docs = docs
        self.dictionary = dictionary

    def __iter__(self):
        for doc in self.docs:
            yield self.dictionary.doc2bow(doc)

class Indexer():
    def __init__(self, path_docs = "../documents/*/*.txt", dictionary = None,
        bow = None):
        self.pipeline = LemmatizerTagger("es_dep_news_trf",
            StopwordsRemover("../resources/spanish_stopwords.txt",
                Tokenizer(SymbolRemover())))
        self.repository = [open(doc, "r", encoding = "utf-8").read()
            for doc in glob.glob(path_docs)]
        if not (dictionary and bow):
            self.documents = Parallel(n_jobs = 12, verbose = 50)(
                delayed(self._preprocess_doc)(doc) for doc in glob.glob(path_docs))
        if dictionary:
            self.dictionary = corpora.Dictionary.load(dictionary)
        if bow:
            self.bow = corpora.MmCorpus(bow)

    def index(self):
        """
        Crea el diccionario de la coleccion y la bolsa de palabras.
        """
        self.dictionary = self._create_dictionary()
        self.dictionary.save("../resources/dictionary.dict")
        self.bow = MyCorpus(self.documents, self.dictionary)
        corpora.MmCorpus.serialize("../resources/bow.mm", self.bow, metadata=True)

    def _create_dictionary(self):
        """
        Funcion especifica encargada de crear y guardar el diccionario.
        """
        dictionary = corpora.Dictionary(doc for doc in self.documents)
        return dictionary

    def _preprocess_doc(self, doc):
        with open(doc, "r", encoding = "utf-8") as f:
            content = f.read()
        return [token["lemma"] for token in self.pipeline.execute(content)]
```

### 6.2.3. Módulo de traducción

El módulo de traducción es el que contiene toda la funcionalidad necesaria para traducir la *query* en inglés al español, mediante el uso de la interlingua UNL.

**Diccionario UNL** Como comentamos en la Sección 5.2, haremos uso de diccionarios para traducir desde el inglés a UNL y después traducir de UNL a español. A pesar de la buena representación que ofrece UNL para las palabras, los diccionarios existentes en UNL son muy escasos y crear uno específico para este experimento llevaría mucho tiempo. Es por ello que decidimos usar el diccionario "*Universal Dictionary of Concepts*" [14], que a pesar de no ser de gran calidad, nos permite de una manera rápida realizar traducciones de inglés a español a través de UNL.

A partir del diccionario inglés-UNL, hemos creado una simplificación en la que únicamente incluimos para el inglés la palabra lematizada, su categoría morfológica y su correspondiente palabra universal en UNL (Figura 6.4.

| | lemma | uw | pos |
|---|---|---|---|
| 0 | 1 chronicles | 1_chronicles(iof>sacred_text>information) | n |
| 1 | 1 kings | 1_kings(iof>sacred_text>information) | n |
| 2 | 1 samuel | 1_samuel(iof>sacred_text>information) | n |
| 3 | 24 hours | 24_hours(icl>period>time,icl>unit) | n |
| 4 | 24-karat gold | 24-karat_gold(icl>gold>thing) | n |

Figura 6.4: Creación del diccionario inglés-UNL.

De la misma forma, a partir del diccionario español-UNL, hemos creado una simplificación en la que en este caso incluimos la representación en UNL y su correspondiente palabra en español.

Por tanto, en la Figura 6.5 podemos ver cómo quedaría el módulo de traducción por dentro en el caso de que se introdujese una *query* en inglés.

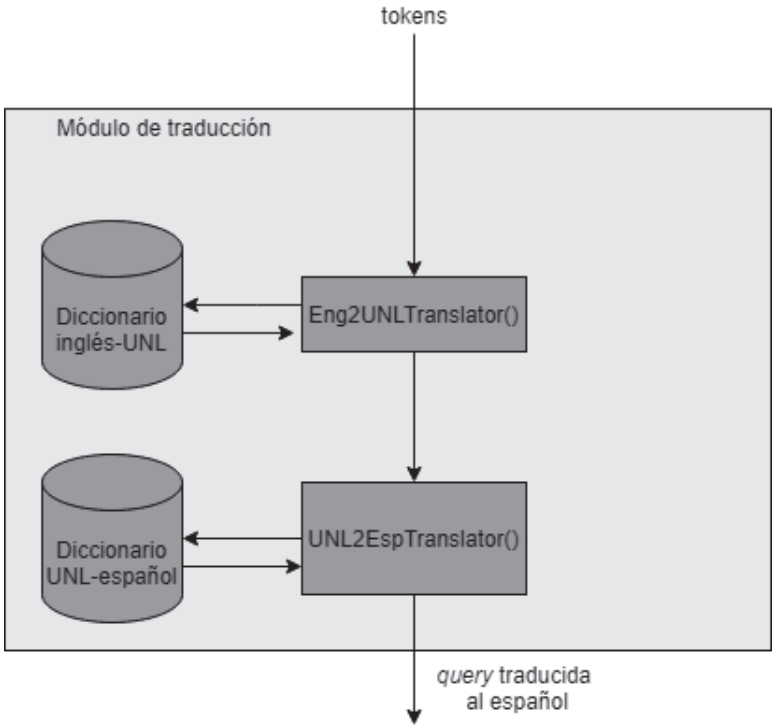



Figura 6.5: Módulo de traducción.

Es importante tener en cuenta que si una palabra determinada no tiene una entrada en el diccionario o si la representación en UNL no tiene entrada en el diccionario UNL-español, esa palabra quedaría sin traducir.

### 6.2.4. Módulo de búsqueda

Finalmente, el módulo de búsqueda contiene la funcionalidad encargada de calcular la similitud entre la *query* introducida por el usuario una vez preprocesada y traducida (en caso de haberse formulado en inglés) y los documentos en español.

Para la implementación de este módulo hemos usado la clase **Similarity** del módulo *similarities* de la librería **gensim**, para realizar el cálculo de similitud de una manera rápida y eficiente entre la *query* del usuario y los documentos. Además, se ha usado también el **modelo Tfidf** de gensim para sacar las características numéricas de los documentos y las *queries*.

A continuación podemos ver el código de la implementación de la función de búsqueda, donde en función de la lengua de la *query* se traduce o no, y después se calcula la similitud con los documentos y se muestra el inicio de los 5 documentos más relevantes.

```
def search(self, lang, query, k = 100, verbose = True):
    """
    En funcion del lenguaje, ejecuta una pipeline u otra y realiza la busqueda.
    """
    if lang == "eng":
        pq = self.traductor.translate(self.pipeline_eng.execute(query))
    elif lang == "esp":
        pq = [token["lemma"] for token in self.pipeline_esp.execute(query)]

    vq = self.dictionary.doc2bow(pq)
    qtfidf = self.model[vq]
    sim = self.index[qtfidf]
    ranking = sorted(enumerate(sim), key=itemgetter(1), reverse=True)
    if verbose:
        print(f"Query ==> {pq}")
        for doc, score in ranking[:3]:
            print("[ Score = " + "%.3f" % round(score,3) + " ] " +
                self.documents[doc][:k])
```

## 6.3. Análisis de resultados

Finalmente, en esta sección presentaremos los resultados obtenidos en los diferentes experimentos realizados.

### 6.3.1. Experimento 1: Búsqueda monolingüe vs búsqueda multilingüe

Con este primer experimento lo que se está midiendo es cuánto empeora nuestro sistema al realizar las búsquedas de documentos en español desde el inglés. Como ya se comentó en el diseño de la experimentación, estamos midiendo la **P@3**, por lo que los posibles valores que tomará cada ejecución serán:

- **0** si ninguno de los 3 documentos tiene que ver con la *query*.
- **0,33** si 1 de los documentos es relevante para la *query*.
- **0,66** si 2 de los 3 documentos son relevantes.
- **1** si todos los documentos son relevantes.

Para ello, hemos realizado primero una búsqueda con las 75 consultas en español (ESP2ESP). Los resultados de esta primera búsqueda se pueden comprobar en la Figura 6.6.

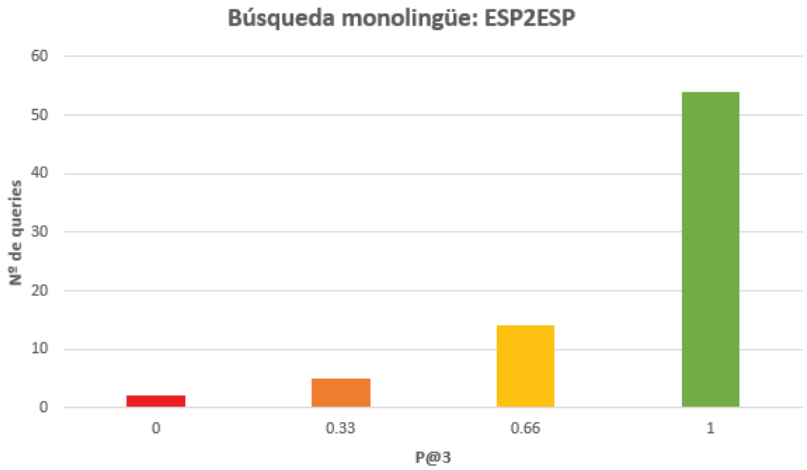



Figura 6.6: Experimento Nº1 - ESP2ESP.

En esta figura podemos observar que en 54 de las 75 *queries* realizadas los 3 documentos mostrados eran relevantes, en 14 *queries* 2 documentos eran relevantes, 5 en las que solamente un documento era relevante y 2 en las que ningún documento mostrado era relevante.

A continuación, ejecutamos el mismo grupo de 75 *queries* pero esta vez en inglés. El resultado de esta ejecución es prácticamente opuesto al anterior (véase Figura 6.7).

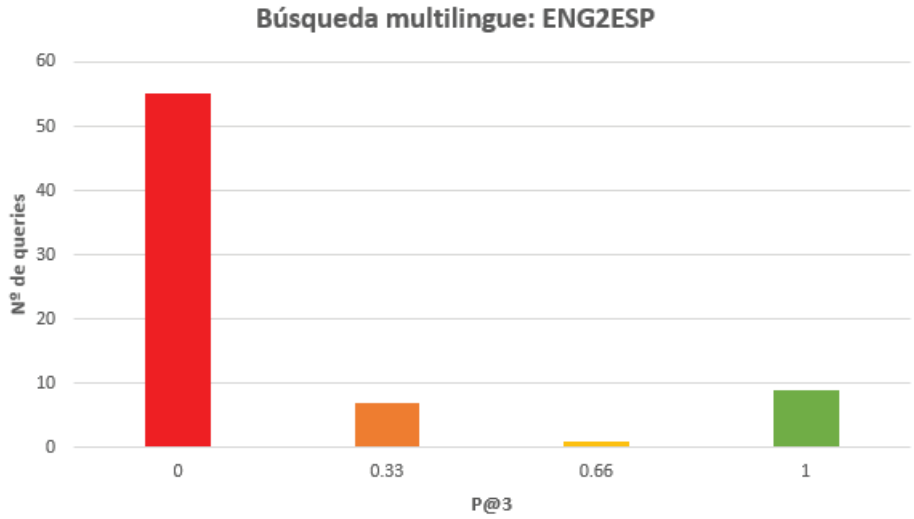



Figura 6.7: Experimento Nº1 - ENG2ESP.

En esta ejecución tan solo en 9 de las 75 *queries* los 3 documentos mostrados eran relevantes. En la Tabla 6.2, se puede observar cómo han cambiado los resultados a peor en la ejecución ENG2ESP respecto a la ejecución ESP2ESP. Es especialmente curioso ver cómo en 45 *queries* en las que se obtenían 3 documentos relevantes buscando desde español ahora ya no se obtienen los mismos resultados e, inversamente, las *queries* en las que no se obtienen ningún documento relevante aumentan en 53.

Estos malos resultados en la ejecución ENG2ESP no pueden ser debidos al buscador, ya que si no también se obtendrían malos resultados en el experimento ESP2ESP. Por lo tanto, estos malos resultados pueden ser debidos al lematizador de inglés o a la calidad del diccionario ENG-UNL o UNL-ESP

Tabla 6.2: Experimento Nº1 - ESP2ESP vs ENG2ESP

| P@3 | 0 | 0.33 | 0.66 | 1 |
|---|---|---|---|---|
| **ESP2ESP** | 2 | 5 | 14 | 54 |
| **ENG2ESP** | 55 | 7 | 1 | 9 |
| **Variación** | +53 | +2 | -13 | -45 |

empleado.

### 6.3.2. Experimento 2: Calidad de la traducción

Debido a los malos resultados obtenidos en el experimento ENG2ESP, se realizó otro experimento en el que se comprobó la calidad de la traducción obtenida por nuestro sistema. Para evaluar este experimento, se han utilizado los siguientes valores:

- **Buena**: cuando la traducción de la *query* es exacta o contiene algún sinónimo.
- **Regular**: cuando la traducción de la *query* no transmite al 100 % la semántica original.
- **Mala**: cuando la traducción es errónea.
- **Nula**: cuando no se consigue traducir una *query*.

En la Figura 6.8 podemos ver los resultados de este experimento, pudiendo observar la mala calidad de las traducciones ya que en 39 de las 75 *queries* ni si quiera se consigue realizar una traducción. Además, tan solo en 11 *queries* se obtiene una calidad de traducción ”buena”.

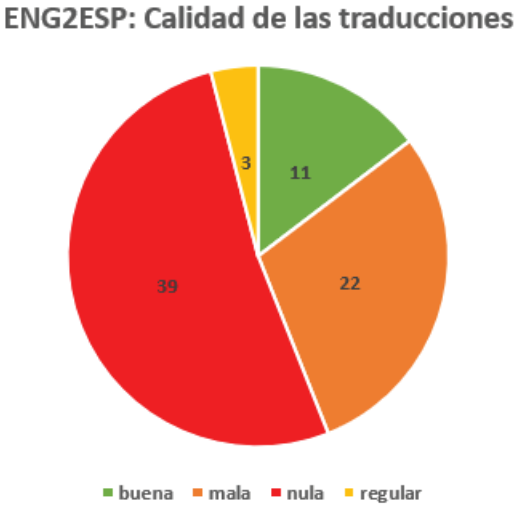



Figura 6.8: Experimento Nº2 - Calidad de las traducciones.

### 6.3.3. Experimento 3: Búsqueda multilingüe vs búsqueda multilingüe refinada

Como se comprobó en el experimento anterior, la calidad de las traducciones obtenidas es muy mala debido al uso de un diccionario ya existente, el cual no ha sido creado de una manera correcta. La calidad del diccionario influye de manera directa en la calidad de la traducción y por tanto también en la búsqueda posterior. Para demostrar esta hipótesis se ha realizado un refinado manual del

diccionario incorporando entradas con el fin de demostrar que, si el diccionario es ”bueno”, nuestro modelo obtiene buenos resultados.

Para ello, se han ejecutado de nuevo las *queries* pero esta vez haciendo uso de los diccionarios refinados. La calidad de las traducciones (véase Figura 6.9) esta vez es muy buena, obteniendo en 62 *queries* una traducción ”buena”.

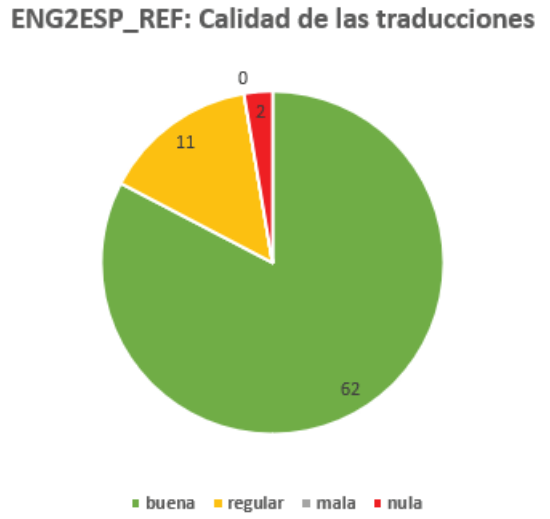



Figura 6.9: Experimento Nº3 - Calidad de las traducciones refinadas.

Esta mejora en la calidad de las traducciones también se convierte en una mejora en la búsqueda. En la Figura 6.10, se muestra una comparación de la ejecución ENG2ESP (azul) versus la ejecución ENG2ESP con los diccionarios refinados (naranja).

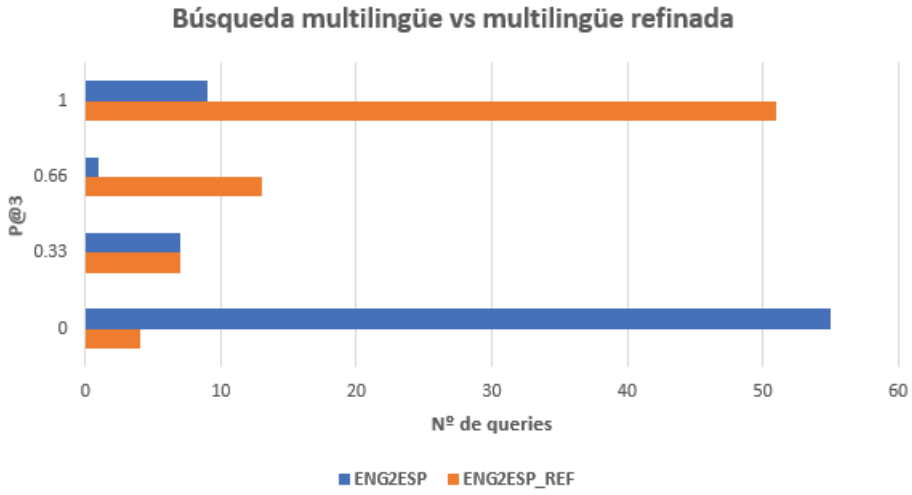



Figura 6.10: Experimento Nº3 - ENG2ESP vs ENG2ESP_REF.

En ella se puede ver cómo en la ejecución ENG2ESP_REF los resultados son muy buenos y mejoran a los de la ejecución ENG2ESP, confirmando así nuestra hipótesis de que la calidad del diccionario influye de manera directa en la calidad de la búsqueda.

Además, a modo de resumen, en la Tabla 6.3 se puede ver la variación en la P@3 de la versión ENG2ESP_REF respecto a la ejecución de referencia ESP2ESP, donde tan solo empeora en 4 de las 75 *queries*.

Tabla 6.3: Experimento Nº3: ESP2ESP vs ENG2ESP_REF

| P@3 | 0 | 0.33 | 0.66 | 1 |
|---|---|---|---|---|
| **ESP2ESP** | 2 | 5 | 14 | 54 |
| **ENG2ESP_REF** | 4 | 7 | 13 | 51 |
| **Variación** | +2 | +2 | -1 | -3 |

# Capítulo 7

# Conclusiones y trabajos futuros

Como se ha visto en el capítulo anterior, el modelo planteado en esta tesis fin de máster obtiene buenos resultados al realizar una búsqueda multilingüe siempre y parece cumplir con los objetivos de este trabajo.

- El enfoque de usar una interlingua, en nuestro caso UNL, para resolver el problema de la multilingualidad en los buscadores documentales parece ser una buena forma de superar la barrera del lenguaje en este área, tal como se ha podido comprobar en la ejecución *ENG2ESP_REF*.
- Sin embargo, se ha podido comprobar que el uso de una interlingua de por sí no garantiza la obtención de buenos resultados (ejecución *ENG2ESP*). Los recursos léxicos empleados, como los diccionarios en este trabajo, deberán haber sido creados con mucho cuidado y de manera precisa ya que de la calidad de estos recursos dependerá la calidad de la posterior búsqueda multilingüe.

  Este hecho podría decirse que es la principal desventaja al poder suponer un cuello de botella a la hora de realizar las búsquedas multilingües. Por el contrario, la ventaja es que estos recursos se construyen con conocimiento del dominio por parte de lingüistas y filólogos y no a partir de datos, como pueden ser otros enfoques como la traducción automática, siendo así sencillo el empleo de este modelo con lenguajes exóticos y poco investigados.
- Además, el empleo de este modelo en ambientes verdaderamente multilingües resultaría una tarea sencilla y escalable, siendo un buen enfoque para afrontar la multilingualidad masiva.
- Finalmente, es de destacar que, a pesar de que las principales líneas de investigación en los últimos años se centran en el uso de técnicas de aprendizaje profundo para resolver la tarea propuesta en este trabajo, el uso de técnicas más clasicas como es en este caso una interlingua supone la obtención de buenos resultados de una manera mucho más transparente y explicable.

Como principales líneas futuras de este trabajo, se plantean los siguientes puntos:

1. Sería interesante incorporar el uso de algún otro recurso léxico, como los tesauros, con los que realizar expansión de la *query* aumentando así la precisión y la cobertura de nuestro sistema.
2. Además de alguna técnica de expansión de la *query*, el uso de una técnica más elaborada de desambiguación semántica a la hora de seleccionar los términos en el diccionario mejoraría con creces la calidad de las traducciones y por tanto de la búsqueda.
3. A la hora de realizar las búsquedas, mejoraría la calidad general del sistema que el usuario

no tuviese que indicar la lengua en la que está realizando la consulta sino que sea el propio sistema el que detectase la lengua empleada por el usuario.

4. Ampliar el tamaño de la colección de documentos empleada para comprobar el funcionamiento de este modelo en un escenario más realistas.
5. Respecto a la escalabilidad en la multilingualidad, habría que indexar los documentos mediante las representaciones de las palabras en UNL para así permitir las búsquedas en el escenario "búsquedas desde varias lenguas de documentos en varias lenguas", siendo este escenario el ideal y más complejo en los buscadores multilingües.
6. Por último, sería necesario revisar los modelos de buscadores y hacer uso de algún método más moderno capaz de capturar la semántica de las palabras, como *Word2Vec*, mejorando así la calidad de las búsquedas.

# Bibliografía


[1] S. Acid, L. M. De Campos, J. M. Fernández-Luna y J. F. Huete, "An information retrieval model based on simple Bayesian networks", *International Journal of Intelligent Systems*, vol. 18, n.º 2, págs. 251-265, 2003. DOI: https://doi.org/10.1002/int.10088.

[2] M. Adriani, "Using Statistical Term Similarity for Sense Disambiguation in Cross-Language Information Retrieval", *Information Retrieval*, vol. 2, págs. 71-82, 2004.

[3] R. Attar y A. S. Fraenkel, "Local Feedback in Full-Text Retrieval Systems", *J. ACM*, vol. 24, n.º 3, págs. 397-417, jul. de 1977, ISSN: 0004-5411. DOI: 10.1145/322017.322021.

[4] R. Baeza-Yates y B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley, 2011, ISBN: 9780321416919.

[5] L. Ballesteros y W. B. Croft, "Dictionary methods for cross-lingual information retrieval", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1134 LNCS, págs. 791-801, 1996, ISSN: 16113349. DOI: 10.1007/bfb0034731.

[6] ——, "Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval", en *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '97, Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1997, págs. 84-91, ISBN: 0897918363. DOI: 10.1145/258525.258540.

[7] ——, "Resolving Ambiguity for Cross-Language Retrieval", en *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '98, Melbourne, Australia: Association for Computing Machinery, 1998, págs. 64-71, ISBN: 1581130155. DOI: 10.1145/290941.290958.

[8] R. D. Brown, "Automatically-Extracted Thesauri for Cross-Language IR: When Better is Worse", en *1st Workshop on Computational Terminology (Computerm*, 1998, págs. 15-21.

[9] J. Cardeñosa, C. Gallardo y L. Iraola, "Interlinguas: A Classical Approach for the Semantic Web. A Practical Case", en *MICAI 2006: Advances in Artificial Intelligence*, A. Gelbukh y C. A. Reyes-Garcia, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, págs. 932-942, ISBN: 978-3-540-49058-6.

[10] J. Cardeñosa, C. Gallardo, L. Iraola y M. A. D. la Villa, "A New Knowledge Representation Model to Support Multilingual Ontologies. A case Study", en *Proceedings of the 2008 International Conference on Semantic Web & Web Services, SWWS 2008, July 14-17, 2008, Las Vegas, Nevada, USA*, CSREA Press, 2008, págs. 313-319, ISBN: 1-60132-089-2.

[11] J. Cardeñosa, C. Gallardo y M. A. de la Villa, "Interlingual Information Extraction as a Solution for Multilingual QA Systems", en *Flexible Query Answering Systems*, T. Andreasen, R. R. Yager, H. Bulskov, H. Christiansen y H. L. Larsen, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, págs. 500-511, ISBN: 978-3-642-04957-6.

[12] M. W. Davis, "New Experiments In Cross-Language Text Retrieval At NMSU's Computing Research Lab", en *Proceedings of The Fifth Text REtrieval Conference, TREC 1996, Gaithersburg,*

*Maryland, USA, November 20-22, 1996*, E. M. Voorhees y D. K. Harman, eds., ép. NIST Special Publication, vol. 500-238, National Institute of Standards y Technology (NIST), 1996.

[13] M. W. Davis y W. C. Ogden, "Implementing Cross-Language Text Retrieval Systems for Large-Scale Text Collections and the World Wide Web", inf. téc., 1997, págs. 2-10.

[14] dikonov, *Universal Dictionary of Concepts*, https://github.com/dikonov/Universal-Dictionary-of-Concepts, 2014.

[15] T. Dunning y M. W. Davis, "Multi-lingual information retrieval", Computing Research Laboratory, New Mexico State University, inf. téc., 1993.

[16] D. Eichmann, M. E. Ruiz y P. Srinivasan, "Cross-Language Information Retrieval with the UMLS Metathesaurus", en *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '98, Melbourne, Australia: Association for Computing Machinery, 1998, págs. 72-80, ISBN: 1581130155. DOI: 10.1145/290941.290959.

[17] C. Fellbaum, ed., *WordNet: An Electronic Lexical Database*, ép. Language, Speech, and Communication. Cambridge, MA: MIT Press, 1998, ISBN: 978-0-262-06197-1.

[18] M. Franco-Salvador, P. Rosso y R. Navigli, "A Knowledge-based Representation for Cross-Language Document Retrieval and Categorization", en *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden: Association for Computational Linguistics, abr. de 2014, págs. 414-423. DOI: 10.3115/v1/E14-1044.

[19] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter y K. E. Lochbaum, "Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure", en *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '88, Grenoble, France: Association for Computing Machinery, 1988, págs. 465-480, ISBN: 2706103094. DOI: 10.1145/62437.62487.

[20] A. Gabarre González y C. García Cabeza, "Clasificador Documental con Glosario", Ingeniería Lingüística, Máster Universitario en Inteligencia Artificial, inf. téc., dic. de 2020. dirección: https://github.com/themrcesi/Linguistic-Engineering/blob/main/Documents-Classifier/gabarre_garcia_documents_classifier.pdf.

[21] J. Gao, M. Zhou, J.-Y. Nie, H. He y W. Chen, "Resolving Query Translation Ambiguity Using a Decaying Co-Occurrence Model and Syntactic Dependence Relations", en *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '02, Tampere, Finland: Association for Computing Machinery, 2002, págs. 183-190, ISBN: 1581135610. DOI: 10.1145/564376.564409.

[22] J. Gonzalo, F. Verdejo e I. Chugur, "Using EuroWordNet in a concept-based approach to cross-language text retrieval", *Applied Artificial Intelligence*, vol. 13, n.º 7, págs. 647-678, 1999, ISSN: 08839514. DOI: 10.1080/088395199117234.

[23] *HEREIN System*. dirección: https://www.coe.int/en/web/herein-system/about.

[24] M. Honnibal, I. Montani, S. Van Landeghem y A. Boyd, *spaCy: Industrial-strength Natural Language Processing in Python*, 2020. DOI: 10.5281/zenodo.1212303. dirección: https://doi.org/10.5281/zenodo.1212303.

[25] D. A. Hull y G. Grefenstette, "Querying across Languages: A Dictionary-Based Approach to Multilingual Information Retrieval", en *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '96, Zurich, Switzerland: Association for Computing Machinery, 1996, págs. 49-57, ISBN: 0897917928. DOI: 10.1145/243199.243212.

[26] Z. Jiang, A. El-Jaroudi, W. Hartmann, D. Karakos y L. Zhao, "Cross-lingual Information Retrieval with BERT", English, en *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, Marseille, France: European Language Resources Association, mayo de 2020, págs. 26-31, ISBN: 979-10-95546-55-9.

[27] S. K.Dwivedi y G. Chandra, "A Survey on Cross Language Information Retrieval", *International Journal on Cybernetics & Informatics*, vol. 5, n.º 1, págs. 127-142, feb. de 2016, ISSN: 23208430. DOI: 10.5121/ijci.2016.5113.

[28] K. Kishida, "Technical issues of cross-language information retrieval: A review", *Information Processing and Management*, vol. 41, n.º 3, págs. 433-455, mayo de 2005, ISSN: 03064573. DOI: 10.1016/j.ipm.2004.06.007.

[29] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes y D. Brown, *Text classification algorithms: A survey*, 2019. DOI: 10.3390/info10040150. arXiv: 1904.08067.

[30] G. A. Levow, D. W. Oard y P. Resnik, "Dictionary-based techniques for cross-language information retrieval", *Information Processing and Management*, vol. 41, n.º 3, págs. 523-547, mayo de 2005, ISSN: 03064573. DOI: 10.1016/j.ipm.2004.06.012.

[31] R. Litschko, G. Glavaš, S. P. Ponzetto e I. Vulić, "Unsupervised Cross-Lingual Information Retrieval using Monolingual Data Only", 2018. DOI: 10.1145/nnnnnnn.nnnnnnn. arXiv: 1805.00879v1.

[32] E. Loper y S. Bird, "NLTK: The Natural Language Toolkit", *CoRR*, vol. cs.CL/0205028, 2002.

[33] R. Majumder, A. Berntson, ( D. Jiang, J. Gao, F. Wei y N. Duan, *The science behind semantic search: How AI from Bing is powering Azure Cognitive Search*, mar. de 2021. dirección: https://www.microsoft.com/en-us/research/blog/the-science-behind-semantic-search-how-ai-from-bing-is-powering-azure-cognitive-search/.

[34] C. D. Manning, P. Raghavan y H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008, ISBN: 0521865719.

[35] J. S. McCarley, "Should We Translate the Documents or the Queries in Cross-Language Information Retrieval?", ép. ACL '99, College Park, Maryland: Association for Computational Linguistics, 1999, págs. 208-214, ISBN: 1558606093. DOI: 10.3115/1034678.1034716.

[36] J. R. McDonnell, R. G. Reynolds y D. B. Fogel, "Query Translation Using Evolutionary Programming for Multi-Lingual Information Retrieval", en *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*. 1995, págs. 175-185.

[37] T. Mikolov, K. Chen, G. Corrado y J. Dean, "Efficient Estimation of Word Representations in Vector Space", en *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio e Y. LeCun, eds., 2013.

[38] B. Mitra y N. Craswell, "An Introduction to Neural Information Retrieval", *Foundations and Trends® in Information Retrieval*, vol. 13, n.º 1, págs. 1-126, dic. de 2018.

[39] J. Y. Nie, "Cross-language information retrieval", *Synthesis Lectures on Human Language Technologies*, vol. 3, n.º 1, págs. 1-142, ene. de 2010, ISSN: 19474040. DOI: 10.2200/S00266ED1V01Y201005HLT

[40] D. W. Oard y P. G. Hackett, "Document Translation for Cross-Language Text Retrieval at the University of Maryland", en *The Sixth Text REtrieval Conference (TREC-6)*, 1997.

[41] W. D. Oard, "Alternative Approaches for Cross-Language Text Retrieval", inf. téc., 1997, págs. 154-162.

[42] A. Pirkola, "The Effects of Query Structure and Dictionary Setups in Dictionary-Based Cross-Language Information Retrieval", en *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '98, Melbourne, Australia: Association for Computing Machinery, 1998, págs. 55-63, ISBN: 1581130155. DOI: 10.1145/290941.290957.

[43] A. Pirkola, T. Hedlund, H. Keskustalo y K. Järvelin, "Dictionary-Based Cross-Language Information Retrieval: Problems, Methods, and Research Findings", *Information Retrieval*, vol. 4, n.º 3-4, págs. 209-230, 2001, ISSN: 13864564. DOI: 10.1023/A:1011994105352.

[44] R. Rehurek y P. Sojka, "Gensim–python framework for vector space modelling", *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, n.º 2, 2011.

[45] R. Richardson y A. F. Smeaton, "Using wordnet in a knowledge-based approach to information retrieval", inf. téc., 1995.

[46] S. E. Robertson, "The Probability Ranking Principle in IR", en *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, págs. 281-286, ISBN: 1558604545.

[47] G. Salton, "Automatic Processing of Foreign Language Documents", en *Proceedings of the 1969 Conference on Computational Linguistics*, ép. COLING '69, Sång-Säby, Sweden: Association for Computational Linguistics, 1969, págs. 1-28. DOI: 10.3115/990403.990407.

[48] G. Salton, A. Wong y C. S. Yang, "A Vector Space Model for Automatic Indexing", *Commun. ACM*, vol. 18, n.º 11, págs. 613-620, nov. de 1975, ISSN: 0001-0782. DOI: 10.1145/361219.361220.

[49] G. Salton, "Experiments in Multi-Lingual Information Retrieval", USA, inf. téc., 1972.

[50] ——, *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968, ISBN: 0070544859.

[51] G. Salton y C. Buckley, "Improving Retrieval Performance by Relevance Feedback", en *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, págs. 355-364, ISBN: 1558604545.

[52] H. Schütze y J. O. Pedersen, "A Cooccurrence-Based Thesaurus and Two Applications to Information Retrieval", *Inf. Process. Manage.*, vol. 33, n.º 3, págs. 307-318, mayo de 1997, ISSN: 0306-4573. DOI: 10.1016/S0306-4573(96)00068-4.

[53] *Search Engine Market Share Worldwide*. dirección: https://gs.statcounter.com/search-engine-market-share.

[54] P. Sheridan y J. P. Ballerini, "Experiments in Multilingual Information Retrieval Using the SPIDER System", en *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '96, Zurich, Switzerland: Association for Computing Machinery, 1996, págs. 58-65, ISBN: 0897917928. DOI: 10.1145/243199.243213.

[55] P. Sheridan, M. Braschler y P. Schäuble, "Cross-Language Information Retrieval in a Multilingual Legal Domain", en *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, ép. ECDL '97, Berlin, Heidelberg: Springer-Verlag, 1997, págs. 253-268, ISBN: 3540635548.

[56] D. Soergel, "Multilingual Thesauri in Cross-Language Text and Speech Retrieval", en *In Working Notes of AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, 1997, págs. 164-170.

[57] T. Talvensaari, J. Laurikkala, K. Järvelin, M. Juhola y H. Keskustalo, "Creating and Exploiting a Comparable Corpus in Cross-Language Information Retrieval", *ACM Trans. Inf. Syst.*, vol. 25, n.º 1, 4-es, feb. de 2007, ISSN: 1046-8188. DOI: 10.1145/1198296.1198300.

[58] E. M. Voorhees, "Using WordNet to Disambiguate Word Senses for Text Retrieval", en *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '93, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1993, págs. 171-180, ISBN: 0897916050. DOI: 10.1145/160688.160715.

[59] X. Wei y W. B. Croft, "LDA-Based Document Models for Ad-Hoc Retrieval", en *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '06, Seattle, Washington, USA: Association for Computing Machinery, 2006, págs. 178-185, ISBN: 1595933697. DOI: 10.1145/1148170.1148204.

[60] S. Wiltrud Kessler, "A Multilingual Search Engine Based on the Universal Networking Language", Trabajo Final de Sistemas Informáticos, Facultad de Informática - Universidad Politécnica de Madrid, inf. téc., 2009.

[61] J. Xu y W. B. Croft, "Query Expansion Using Local and Global Document Analysis", en *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ép. SIGIR '96, Zurich, Switzerland: Association for Computing Machinery, 1996, págs. 4-11, ISBN: 0897917928. DOI: 10.1145/243199.243202.

[62] M. C. Yang, W. H. Wood y M. R. Cutkosky, "Design Information Retrieval: A Thesauri-Based Approach for Reuse of Informal Design Information", *Eng. with Comput.*, vol. 21, n.º 2, págs. 177-192, nov. de 2005, ISSN: 0177-0667.

[63] D. Zhou, M. Truran, T. Brailsford, V. Wade y H. Ashman, "Translation Techniques in Cross-Language Information Retrieval", *ACM Comput. Surv.*, vol. 45, n.º 1, dic. de 2012, ISSN: 0360-0300. DOI: 10.1145/2379776.2379777.

[64] Z. Zhu, M. Li, L. Chen y Z. Yang, "Building Comparable Corpora Based on Bilingual LDA Model", en *ACL (2)*, 2013, págs. 278-282.