CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Diseño e implementación de un algoritmo para la detección de la negación de textos clínicos en español

AUTOR: Jian Chen Zhang

DIRECTORA: Ernestina Menasalvas

MADRID, JUNIO DE 2014

AGRADECIMIENTOS

Durante mucho tiempo parecía imposible que fuera capaz de sacarme la carrera. Si finalmete ha sido posible ha sido gracias al apoyo de todas esas personas que me han ayudado a seguir adelante. Muchas gracias.

A *mis padres*, por luchar incansablemente para darme las oportunidades que ellos nunca tuvieron.

A *mi hermana*, por estar siempre ahí.

A *Ernestina*, por confiar en mí y permitirme participar en este gran proyecto.

A *Roberto*, por su inestimable ayuda a lo largo del proyecto.

A *Julián*, porque siempre es genial conocer a alguien con tu mismo sentido del humor.

A *Daniel*, por todos estos años de amistad.

A *Andrés*, por ser tan reperra.

A *Daniel*, porque ahora ya puedo decir que conozco a un youtuber.

A *Sandra y Guillermo*, por todas las terribles prácticas vividas en buena compañía.

A *Pau*, porque nunca sabes cuando vas a conocer a un gran amigo.

A *Jorge*, por ser el más troll de toda la facultad.

A *Natalia*, por su infinita generosidad.

A *Mariló e Irene*, porque siempre tenéis algo nuevo que enseñarme.

A *Ricardo y Marisa*, por enseñarme que hay profesores que realmente sienten pasión por su profesión.

A *ACM*, por los grandes momentos y mejores amigos que me ha dado.

Al *CTB*, y el fantástico ambiente que habéis conseguido.

Al *Bang*, por esas tardes tan épicas.

RESUMEN

Vivimos en una época en la que cada vez existe una mayor cantidad de información. En el dominio de la salud la historia clínica digital ha permitido digitalizar toda la información de los pacientes. Estas historias clínicas digitales contienen una gran cantidad de información valiosa escrita en forma narrativa que sólo podremos extraer recurriendo a técnicas de procesado de lenguaje natural. No obstante, si se quiere realizar búsquedas sobre estos textos es importante analizar que la información relativa a síntomas, enfermedades, tratamientos etc. se puede refererir al propio paciente o a sus antecentes familiares, y que ciertos términos pueden aparecer negados o ser hipotéticos. A pesar de que el español ocupa la segunda posición en el listado de idiomas más hablados con más de 500 millones de hispano hablantes, hasta donde tenemos conocimiento no existe ningún método de detección de la negación, probabilidad e histórico en textos clínicos en español.

Por tanto, este Trabajo Fin de Grado presenta una implementación basada en el algoritmo ConText para la detección de la negación, probabilidad e histórico en textos clínicos escritos en español.

El algoritmo se ha validado con 454 oraciones que incluían un total de 1897 disparadores obteniendo unos resultado de 83.5 %, 96.1 %, 96.9 %, 99.7 % y 93.4 % de exactitud con condiciones afirmados, negados, probable, probable negado e histórico respectivamente.

Palabras clave: Data mining, Text mining, Tecnología biomédica, Historia clínica digital, detección de la negación, probabilidad e histórico.

ABSTRACT

We live in an era in which there is a huge amount of information. In the domain of health, the electronic health record has allowed to digitize all the information of the patients. These electronic health records contain valuable information written in narrative form that can only be extracted using techniques of natural language processing. However, if you want to search on these texts is important to analyze if the relative information about symptoms, diseases, treatments, etc. are referred to the patient or family casework, and that certain terms may appear negated or be hypothesis. Although Spanish is the second spoken language with more than 500 million speakers, there seems to be no method of detection of negation, hypothesis or historical in medical texts written in Spanish.

Thus, this bachelor's final degree presents an implementation based on the ConText algorithm for the detection of negation, hypothesis and historical in medical texts written in Spanish.

The algorithm has been validated with 454 sentences that included a total of 1897 triggers getting a result of 83.5 %, 96.1 %, 96.9 %, 99.7 % and 93.4 % accuracy with affirmed, negated, hypothesis, negated hypothesis and historical respectively.

Keywords: Data mining, Text mining, Biomedical technology, Electronic health record, detection of the negation, hypothesis and historical.

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

CAPÍTULO 1

INTRODUCCIÓN

1.1. Introducción y motivación

Gracias a la cada vez mayor digitalización de toda la información, y especialmente a Internet, cada día podemos acceder a grandes cantidades de información de forma muy sencilla. En la actualidad se generan, almacenan y distribuyen grandes cantidades de información cada segundo. Esto ha provocado que nos veamos expuestos a una sobreinformación, y que obtener información que realmente sea relevante y nos interese sea una tarea cada vez más costosa. De esta manera, la recuperación, gestión, clasificación y tratamiento de la información para extraer datos valiosos ha ido ganando importancia con el paso del tiempo hasta convertirse en algo fundamental.

Provocado por la digitalización de la información, encontramos en el mundo de la tecnología biomédica la Historia Clínica Electrónica (HCE) [1]. Gracias a esta herramienta, todos los datos médicos de todos los pacientes, tanto estructurada como no (síntomas, enfermedades anteriores, alergias, medicación o estadísticas personales como peso y altura) son almacenados de forma digital, permitiendo su acceso de forma mucho más sencilla y directa. Por tanto, el informe clínico de un individuo es una gran fuente de información, tanto para el propio paciente como para los profesionales que le tratan.

En el campo de la salud los avances tecnológicos han permitido recoger información de todo tipo de los pacientes: datos de sensores, imágenes, texto libre, etc. También se genera, almacena y distribuye grandes cantidades de información referente a todos aquellos pacientes que se tratan o han sido tratados por el diferente personal médico. Todos estos datos contienen una gran cantidad de información que permitiría pasar a la medicina basada en la evidencia. No obstante, si bien se genera y almacena la información, no existen herramientas que permitan, ni siquiera en la mayoría de los casos, consultar esta información. Entre los retos para la consulta y análisis de estos datos destacamos: datos heterogéneos (imagen, texto), velocidad de generación de los datos y falta de estandarización.

La adopción de las HCEs es un proceso en auge que se espera que se extienda rapidamente. Mientras que en el año 2011 sólo el 35 % de los hospitales de EE.UU. disponía de HCEs, se prevee que en el año 2016 esta cifra aumente hasta el 95 %. En España se calcula que en la actualidad aproximadamente el 40 % de la población dispone de una HCE [MdS13].

Si nos centramos en la información textual que se encuentra en las historia digital, ya sea en informes médicos o radiológicos, notas clinicas, etc., nos encontramos con grandes cantidades de datos que potencialmente guardan información valiosa. Es necesario tratar dichas notas clínicas, dado que suelen ser muy extensas e incluyen todo tipo de información: anotaciones que contienen gran cantidad de conocimientos relativa al pasado del paciente (que puede ser relevante en el futuro). Además nos encontramos con información que puede estar tanto en forma de hipótesis, afirmada o negada, lo que supone una acentuación del problema de extracción correcta de la información. Según Chapman et al.

[1] Electronic Health Record (EHR)

[CBH+02], la mitad de los términos en un texto médico aparecen negados. Dichas notas están escritas en lenguaje natural, lo que dificulta la tarea de detección, extracción, gestión y clasificación de la información. Es por ello que el campo de la tecnología biomédica ha invertido una gran cantidad de tiempo y recursos en esta cuestión, produciendo decenas de aplicaciones cuya finalidad es indexar, extraer y codificar síntomas obtenidas de notas clínicas almacenadas de forma digital [FC99].

Entre los retos del tratamiento del lenguaje natural destacan: su diversidad, la cantidad de vocabulario a manejar, información contextual que puede modificar el significado de una oración, la gran cantidad de reglas a tener en cuenta, y entre otros, estas carencias se ven motivadas por una falta de estandares que complica enormemente su modelización para obtener un algoritmo universal.

En este TFG nos focalizamos en el problema de encontrar en un documento escrito en español aquellos términos que están negados, que son hipotéticos o que son históricos. Entre los algoritmos mas representativos para solventar este problema en textos en inglés destaca el algoritmo NeGex, presentado en [CBH+01] y que está diseñado para detectar negación en textos clínicos. Basándose en notas clínicas escritas en lenguaje natural, NegEx determina la presencia o ausencia de síntomas o enfermedades. Para lograrlo, el algoritmo hace uso de expresiones regulares que determinan el alcance de los disparadores.

En [CHV+13] podemos consultar el análisis de los retos en la traducción de los disparadores de negación utilizados para el algoritmo NegEx a otros idiomas como francés, alemán o sueco.

Skeppstedt [Ske11], [SDN11], ha adaptado al sueco las reglas del algoritmo de la detección de la negación NegEx basadas en el idioma inglés. Los resultados obtenidos [Ske11] mostraron una menor precisión y fiabilidad respecto a los valores obtenidos para inglés.

La versión ampliada de este algoritmo, llamada ConText [HDTC09], está basada en expresiones regulares y es capaz de determinar no solo si una condición médica está negada, sino también si se trata de una hipótesis, si hace referecia al historial del paciente o si está relacionado con otra persona que no sea el propio paciente.

Por otro lado, se han propuesto diferentes metodologías basadas en técnicas de machine learning [MLD08], [MD09], [RRM08], uno de ellos, constituido por dos clasificadores que determinan el alcance de la negación en textos biomédicos, se presenta en [MLD08]. Dichos clasificadores determinan si los tokens de una oración muestran signos de negación y buscan el alcance total de dichos tokens respectivamente. Gracias a este método se consiguió reducir el error en un 32.07 % [MD09] respecto a otros sistemas en diferentes tipos de textos.

A pesar de que el español ocupa la segunda posición en el listado de idiomas más hablados con más de 500 millones de hispano hablantes [Cer13], hasta donde tenemos conocimiento no existe ningún método de detección de la negación en textos clínicos en español. Dada la gran cantidad de hispano hablantes, su importancia como idioma y

la cantidad de información valiosa que es generada en dicho idioma cada día, creemos especialmente interesante adaptar algunos de los métodos y algoritmos presentados a lo largo de la introducción al español. Conseguirlo nos permitiría ampliar enormemente la cantidad de información biomédica disponible, dar acceso a sus beneficios a una gran parte de la población (España, América central y Sudamérica) y crear nuevas vías de desarrollo tanto para investigadores como para profesionales médicos.

Consecuentemente, este TFG presenta el diseño, implementación validación y evaluación de un algoritmo basado en ConText que permite detectar la negación, la hipotesis y los hechos históricos en los textos de una historia clínica digital.

1.2. Objetivos

El objetivo principal de este TFG es la implementación de un algoritmo basado en ConText, adaptado para la detección de la negación, hipótesis e información contextual en textos clínicos escritos en español.

Este objetivo global se desglosa en los siguientes objetivos parciales:

1. Establecer los requisitos funcionales de un módulo para la estructuración de notas clínicas (MENC).
2. Diseñar una arquitectura del MENC.
3. Implementar cada uno de los submódulos que integran la arquitectura del MENC.
4. Implementar el MENC.
5. Evaluación y validación el MENC.
6. Generar documentación para usuarios y futuros desarrolladores.

CAPÍTULO 2

ESTADO DE LA CUESTIÓN

2.1. Introducción

Este trabajo tiene por objetivo desarrollar una implementación del algoritmo ConText capaz de detectar y etiquetar correctamente negación, hipótesis e historial en notas clínicas escritas en español. Por tanto es este capítulo vamos a realizar un estudio de los diferentes trabajos previos desarrollados y mostraremos el estado de la cuestión relacionado con la detección de la negación en notas clínicas, los diferentes métodos y algoritmos existentes y su situación actual tanto en español como en diferentes idiomas como inglés, alemán o sueco.

En notas clínicas es fundamental poder saber si un determinado hecho (enfermedad, tratamiento, síntoma, etc.) se refiere al sujeto del que se realiza la nota clínica, si por el contrario se refiere a otro sujeto, si es un hecho afirmado o negado o si solo se trata de una hípotesis; y ser capaces de detectar, extraer, gestionar y clasificar la información. Adicionalmente las notas clínicas están escritas en lenguaje natural, lo que dificulta aún más su procesado. Por tanto necesitamos métodos capaces de trabajar con lenguaje natural y que nos faciliten este proceso.

En [CBH$^+$02] se afirma que la mitad de los términos en un texto médico aparecen negados. Por tanto, ser capaces de detectar correctamente aquellos términos que aparecen de forma negada en una nota clínica, y el alcance de dicha negación, es un paso imprescindible para poder extraer correctamente conocimiento de las mismas.

Dos de los algoritmos más extendidos y utilizados actualmente son NegEx [CBH$^+$01] y ConText [HDTC09], aunque existen diferentes aproximaciones al problema como los propuestos en [CBH$^+$02], [HDTC09], [EBB$^+$05] [MDN01], [HL07], [Ske11], [SDN11] o [RRM08].

Elkin et al. [EBB$^+$05] nos presentan un método basado en NegEx para identificar el alcance de los disparadores de negación. Este sistema se aplica a pequeños fragmentos obtenidos al aislar frases de notas clínicas. Los valores de precisión y exactitud obtenidos son del 91.2 % y 97.2 % respectivamente.

Otra aproximación es Negfinder [MDN01]. Este programa es capaz de identificar patrones de negación presentes en documentos clínicos. Para ello, los documentos son preprocesados sustituyendo palabras clave por un procedimiento capaz de identificar patrones de negación mediante un proceso que utiliza reglas gramaticales. Gracias a esto podemos distinguir la negación y podemos asociarlas con uno o varios conceptos que le precedan o sucedan. La sensibilidad y especificidad obtenidas con Negfinder al ser aplicado a notas clínicas para detectar la negación se encuentran entre el 91 % y el 96 %.

Una solución híbrida se encuentra en [HL07]. Esta aproximación combina expresiones regulares con análisis gramatical para detectar automáticamente negaciones en informes radiológicos. Las oraciones negadas fueron identificadas con una sensibilidad y precisión del 92.6 % y 98.6 % respectivamente.

En [NPCD12] nos presentan un sistema basado en técnicas de Machine Learning que

es capaz de identificar negación y probabilidad en textos clínicos. El sistema que proponen basa su funcionamiento en dos fases: un clasificador decide si cada palabra de la oración es una negación/probabilidad o no y comunica estos resultados a otro clasificador que determina el alcance de las etiquetas detectadas previamente. Para probarlo se recurrió a artículos médicos, resúmenes científicos e informes clínicos, y los resultados obtenidos se compararon con un sistema de detección de la negación basado en expresiones regulares y otro basado en Machine Learning. La precisión obtenida en la detección de la negación fue de un 97.3 % y en la detección de la probabilidad de un 94.9 %. Los falsos positivos se evitaron en un 93.2 % y en un 80.9 % en negación y probabilidad respectivamente y el alcance se obtuvo correctamente en un 90.9 % en el caso de la negación y en un 71.9 % en el caso de la probabilidad. Estos resultados demostraron ser superiores a los obtenidos en los sistemas comparados.

A continuación nos centraremos en los métodos NegEx y ConText ya que el algoritmo que proponemos como solución en este TFG se basa en ellos.

2.2. NegEx

NegEx es un método presentado en [CBH+01] diseñado para detectar la presencia o ausencia de una enfermedad mediante la detección de la negación en textos clínicos escritos en lenguaje natural. El algoritmo utiliza expresiones regulares para determinar la existencia de disparadores que indiquen negación y calcular el alcance de dichos disparadores, etiquetando como negación todo aquello que se encuentre dentro del mismo. NegEx se ha utilizado para detectar negación en informes de alta de pacientes con un 94.5 % de especificidad, 84.5 % de precisión y 78 % de sensibilidad. Una implementación funcional del algoritmo para textos clínicos escritos en inglés desarrollado en el lenguaje de programación Python se puede consultar en [WWC13].

Para detectar la negación se determinan tres tipos distintos de términos que pueden indicar la existencia de la negación:

- Términos pseudo-negados: frases que parecen negadas, pero en las cuales la condición médica no se presenta negada.
- Términos negados en pre-condición: el término que provoca la negación aparece antes de la oración que niega.
- Términos negados en post-condición: el término que provoca la negación aparece después de la oración que niega.

Al igual que existen términos que determinan negación, existen otro tipo de términos llamados terminales. Su función es la de indicar que el alcance de un término negado finaliza. Ej: “El paciente niega dolor torácico pero continúa con insuficiencia respiratoria.” En este caso el término ‘pero’ actúa como terminal.

Para determinar el alcance de los términos se recurre a expresiones regulares que detectan dos posibles casos.

- <término negado> * <término negado|fin de oración>
- <término indexado> * <oración negada>

Siendo * un número determinado o no de palabras que forman parte de una oración.

Para cada una de las oraciones de la nota clínica, el algoritmo busca todos los términos de negación, selecciona la primera de ellas y, en el caso de que sea una pseudo-negación, pasa al siguiente. En el caso de que sea una pre-condición calcula el alcance del término buscando un término terminal, otra negación o pseudo-negación, o alcanzando el final de la oración. Si se trata de una post condición el alcance se calcula en sentido contrario, basándonos en el valor del siguiente disparador al seleccionado.

Originalmente su funcionamiento estaba centrado en notas clínicas escritas en inglés, pero posteriormente fue adaptado a otros idiomas como alemán o sueco.

En [CBH+02] se analiza el rendimiento de NegEx en diferentes tipos de notas clínicas. El estudio utiliza tres tipos distintos de oraciones negadas: pre-condición, post-condición y pseudo-condición. Las oraciones negadas fueron extraídas de diferentes informes de altas médicas previamente analizadas, de un sistema llamado SymText [Sym] y de oraciones negadas añadidas por los propios autores. El resultado obtenido fue de una precisión de media del 97 %. Sin embargo, su rango varía desde un 84 % hasta un 19 % dependiendo de la sección de la nota clínica.

Skeppstedt [Ske11], [SDN11] ha adaptado el algoritmo NegEx al lenguaje sueco. En [Ske11] muestran los resultados obtenidos, siendo estos menos precisos y fiables que los obtenidos por el algoritmo en inglés: 75.2 % de precisión y 81.9 % de sensibilidad en sueco frente a 84.5 % de precision y 78 % de sensibilidada en inglés. Como se presenta en [SDN11], el comportamiento de NegEx se analizó mediante diferentes textos clínicos obtenidos de la Stockholm EPR. En concreto el estudio se centró en términos de SNOMED CT [Org] (terminología médica multilenguaje que nos facilitan los registros médicos electrónicos, lo que nos proporciona un acceso efectivo a la información) bajo la categoría “hallazgos” o “trastornos”. Adicionalmente, se realizó otro estudio presentado en [MS14], donde las notas clínicas escritas en sueco se han analizando enfatizando la búsqueda de cuatro términos: desorden, encontrando, medicación y estructura corporal. Dicho estudio analizó el rendimiento de diferentes métodos de reconocimiento de entidades en notas clínicas escritas en sueco y si dividir las diferentes categorías médicas en entidades más específicas mejoraría sus resultados.

2.3. ConText

ConText [HDTC09], es una versión ampliada del método NegEx capaz de determinar no solo si las condiciones médicas mencionadas en una nota clínica se encuentran nega-

das, sino además de si se trata de una hipótesis, de un hecho histórico o si se refiere a alguien que no es el propio paciente. El algoritmo infiere el tipo de oración gracias a los indicadores gramaticales que se encuentran en el contexto de una condición médica.

Para conseguirlo asigna valores por defecto a cada una de las opciones (negación, temporalidad y sujeto) y los modifica en el caso de encontrar un disparador que lo provoque. En el caso de la negación el valor por defecto es *afirmado*, que modificará en el caso de detectar una negación en la oración.
Ej: "El paciente niega nauseas." Nauseas aparece negado. Por tanto su valor se modificará de afirmado a negado.

Para los valores de temporalidad su valor por defecto es *reciente*. Al igual que el caso anterior su valor se podrá ver modificado en el caso de que sea necesario por los valores *histórico* ó *hipotético*. Aquellas oraciones en las cuales la ventana temporal se encuentre por encima de las dos semanas serán etiquetadas como histórico en vez de como reciente. Si no se puede considerar reciente ni histórico entonces será etiquetada como hipotético.
Ej: "El paciente deberá volver si muestra síntomas de fiebre" será considerado como hipotético ya que en ningún momento se especifica una ventana temporal.

Finalmente, para sujeto, el valor por defecto es *paciente*, pudiendo verse modificado por *otro* si algún disparador en una oración así lo indicara.
Ej: "El padre del paciente tiene un historial de diabetes". Diabetes no afectaría al propio paciente, por lo que sería clasificado como otro.

Al igual que el algoritmo NegEx, ConText clasifica los diferentes términos, en este caso como disparador, pseudo-disparador y terminal. En la versión del algoritmo en inglés se han detectado y clasificado como "disparador" 143 términos negados, 10 históricos, 11 hipotéticos y 26 como otros. En el caso de "pseudo-disparador" se han detectado 17 términos negados, 17 históricos, 4 hipotéticos y 18 como otros. Para los "terminales" la clasificación utilizada para los diferentes términos ha sido la siguiente: Presentation, Patient, Because, Diagnosis, ED, Etiology, Recent, Remain, Consistent, Which, And, y But.

El algoritmo recibe como entrada una oración extraída de una nota clínica en la cual se han indexado las condiciones médicas y se han asignado los valores por defecto a aquellos fragmentos de la oración afectados por un disparador. Como salida obtendremos la oración con los valores correctamente asignados acorde con los resultados obtenidos en el siguiente algoritmo:

1. Marcar todos los términos disparadores, pseudo-disparadores y terminales de la oración.
2. Iterar por los diferentes elementos de la oración de izquierda a derecha.
   - Si el elemento a analizar es un pseudo-disparador, avanzar al siguiente.
   - En otro caso, determinar el alcance del disparador y asignarle los valores contextuales indicados dentro de los términos situados dentro de dicho alcance.

En un estudio inicial [CW07] se muestran los resultados obtenidos por ConText en informes de los servicios de urgencias, con una alta especificidad y precisión para la negación (97 %, 97 % en ambos casos), moderada para temporalidad (hipotético) (83 %, 94 %) y suficiente para temporalidad (histórico) (67 %, 74 %) y sujeto (50 %, 100 %).

ConText no es capaz de detectar condiciones históricas con tanta precisión como condiciones negadas porque existen diferencias significativas en la forma en la que ambos se expresan en una nota clínica:

- La palabra "historial", que es el principal disparador de términos históricos, puede tener diferentes significados e interpretaciones no necesariamente relacionadas con históricos.
- Los términos históricos dependen de su situación y contexto en relación con el resto de la oración.
- El hecho de que un término sea considerado histórico o reciente puede depender de su relación temporal con otras condiciones o eventos de la oración.
- Existen oraciones en las cuales pueden presentarse términos históricos sin ningún disparador claro que lo indique.

Por tanto, podemos determinar que su funcionamiento a la hora de detectar la negación es lo suficientemente bueno como para sustituir a NegEx (en negación en informes de alta de pacientes NegEx obtuvo un 94.5 % de especificidad y un 84.5 % de precisión y Context un 89 % de especificidad y un 84 % de precisión) y adicionalmente nos permite detectar otros términos como histórico o hipótesis. La especificidad y precisión alcanzada son lo suficientemente buenas como para poder aceptarlas pero existe margen de mejora.

CAPÍTULO 3

PLANTEAMIENTO DEL PROBLEMA

3.1. Planteamiento

Del estado de la cuestión presentado se desprende que si bien existen algoritmos para la detección de la negación, hipotesis e histórico en textos clínicos para algunos idiomas como inglés, sueco o alemán, no existen estos algoritmos para el español. En este trabajo planteamos el diseño de este algoritmo basándonos en el algoritmo ConText que se ha presentado en el estado de la cuestión. A continuación detallamos los pasos del proceso que será necesario realizar y terminamos con un análisis de requisitos que nos permitirá diseñar la solución propuesta.

El principal escollo en la extracción de información de las notas clínicas es el hecho de que estén escritas en lenguaje natural. Esto dificulta sobremanera la detección y gestión de la información ya que en la actualidad no existe ninguna forma automatizada de tratar con texto en lenguaje natural, y la información ya obtenida previamente se encuentra en otros idiomas distintos al español.

3.1.1. Listado de posibles disparadores en notas médicas

Antes de poder desarrollar un algoritmo capaz de detectar la negación necesitábamos ser capaces de determinar qué palabras indican la negación, histórico, hipotético de un término, o si dicho término afecta al propio paciente o a otro sujeto (familiar, amigo, compañero de piso, etc.). En español la negación se determina mediante adverbios (no, nunca, nada, etc.), verbos (negar, rehusar) o locuciones (en mi vida). Para nuestro algoritmo nos centraremos en aquellas palabras que se presentan en contextos médicos.

3.1.2. Nomenclatura del etiquetado

Necesitábamos definir un sistema de etiquetado sencillo que nos permitiera mostrar de forma directa y visual si un término se encuentra afirmado, negado, es una hipótesis, etc. y el alcance de dicha afirmación, negación, hipótesis, etc.

3.1.3. Alcance del etiquetado

En una misma oración es habitual encontrar múltiples términos médicos que pueden encontrarse tanto afirmados, como negados, ser hipótesis, referirse a síntomas en pasado, etc. Por tanto es necesario que seamos capaces de determinar el alcance de los diferentes disparadores para que solo etiquete los términos clínicos a los que afecta.

3.1.4. Detección de disparadores en notas médicas

Para poder etiquetar términos primero tenemos que ser capaces de detectar los diferentes disparadores presentes en las notas clínicas que indican afirmación, negación, hipótesis, etc. evitando falsos positivos.

3.1.5. Sentido del etiquetado e incompatiblidades entre las mismas

No todos los disparadores afectan al contenido inmediatamente posterior. Hay que tener en cuenta que un disparador puede afectar tanto a los términos que se encuentran a su derecha ("El paciente muestra inflamación y dolor.") como a los términos situados a su izquierda ("Fiebre ausente."). Además pueden presentarse conflictos (dos disparadores diferentes que etiquetan los mismos términos), por lo que necesitamos asignar diferentes prioridades.

3.1.6. Etiquetado múltiple

El lenguaje natural es ambiguo y permite que una misma palabra adquiera diferentes significados según el contexto de la oración. Esto provocará que haya situaciones en las que tendremos que asignar varias etiquetas a un mismo término clínico.

3.1.7. Sentido múltiple

Al igual que una palabra puede tener diferentes significados, también puede afectar a los elementos situados a su derecha (pre) o a los elementos situados a su izquierda (post) según el contexto y estructura de la oración.

3.1.8. Disparadores compuestos

No todos los disparadores son palabras simples (no, sin, etc). Existen una gran cantidad de disparadores compuestos que además pueden ser muy similares entre sí o incluir disparadores simples. Ej: "con antecedentes de" es un disparador compuesto que determina que un término médico es de tipo histórico, e incluye la palabra 'con', disparador sencillo que implica existencia.

3.1.9. Signo de interrogación

No todas las oraciones tienen que estar escritas de forma afirmada o negada. Se pueden presentar oraciones en forma interrogativa medianto signos de interrogación, lo que modificaría el significado original de los disparadores por el valor 'hipótesis'.

3.1.10. Disparadores "pegajosos"

En español se utilizan conjunciones para enlazar las palabras. Algunas de ellas, como 'y' u 'o' pueden ampliar el alcance de un disparador o provocar un cambio de contexto según el significado de la oración.

3.1.11. Inserción de etiquetas

Una vez detectado el disparador tenemos que insertar las etiquetas en la oración mostrando de forma clara su tipo y alcance sin afectar a su legibilidad ni significado original.

3.1.12. Negación de disparadores

La detección de los disparadores nos permite etiquetar los términos médicos, pero incluir todos los disparadores positivos y su equievalente de forma negada no es una solución viable. Realizarlo así añadiría una gran cantidad de disparadores y sería una solución muy poco fiable ya que solo detectaría los casos contemplados previamente. Por tanto necesitaremos un algoritmo que sea capaz de detectar la negación lo suficientemente inteligente como para no depender del listado de disparadores.

3.2. Especificación de requisitos

Para seguir una correcta línea de desarrollo se ha generado un listado de requisitos funcionales y no funcionales a cumplir para garantizar el correcto funcionamiento del algoritmo.

3.2.1. Requisitos funcionales

Id requisito	RF.1
Nombre requisito	Obtención de disparadores.
Características	Obtener los disparadores que producen afirmación, negación, probabilidad, etc. en notas clínicas.
Descripción requirimiento	Necesitamos un listado de disparadores para poder detectarlos en las oraciones a etiquetar.
Prioridad	Alta

Tabla 3.1: Obtención disparadores

Id requisito	RF.2
Nombre requisito	Entrada de notas clínicas.
Características	Asignar como entrada del algoritmo un fichero con las diferentes notas clínicas.
Descripción requirimiento	Necesitamos disponer de una forma sencilla para asignar las notas que queremos etiquetar.
Prioridad	Alta

Tabla 3.2: Entrada de datos

Id requisito	RF.3
Nombre requisito	Salida de notas clínicas ya anotadas.
Características	Devolver las notas clínicas anotadas en un formato legible y sencillo de manejar.
Descripción requirimiento	Facilitar al usuario las notas clínicas una vez etiquetadas.
Prioridad	Alta

Tabla 3.3: Salida de datos

Id requisito	RF.4
Nombre requisito	Detección de términos médicos afirmados.
Características	Ser capaces de detectar cuando un término médico se encuentra afirmado.
Descripción requirimiento	Para poder extraer conocimiento de las notas clínicas debemos detectar correctamente los diferentes términos y etiquetarlos correctamente.
Prioridad	Alta

Tabla 3.4: Detección de afirmación

Id requisito	RF.5
Nombre requisito	Detección de términos médicos negados.
Características	Ser capaces de detectar cuando un término médico se encuentra negado.
Descripción requirimiento	Para poder extraer conocimiento de las notas clínicas debemos detectar correctamente los diferentes términos y etiquetarlos correctamente.
Prioridad	Alta

Tabla 3.5: Detección de negación

Id requisito	RF.6
Nombre requisito	Detección de términos médicos probables.
Características	Ser capaces de detectar cuando existe la posibilidad de que un término médico se encuentre afirmado.
Descripción requirimiento	Para poder extraer conocimiento de las notas clínicas debemos detectar correctamente los diferentes términos y etiquetarlos correctamente.
Prioridad	Alta

Tabla 3.6: Detección de probabilidad

Id requisito	RF.7
Nombre requisito	Detección de términos médicos históricos.
Características	Ser capaces de detectar cuando un término médico se refiere a su historial.
Descripción requirimiento	Para poder extraer conocimiento de las notas clínicas debemos detectar correctamente los diferentes términos y etiquetarlos correctamente.
Prioridad	Alta

Tabla 3.7: Detección de historial

Id requisito	RF.8
Nombre requisito	Detección de términos médicos referidos a otra persona distinta del paciente.
Características	Ser capaces de detectar cuando un término médico no se refiere al paciente sino a otra persona (familiar, amigo, etc.).
Descripción requirimiento	Asignarle síntomas a un paciente cuando no los padece corrompería la información que podamos extraer de él.
Prioridad	Media

Tabla 3.8: Detección de sujetos

Id requisito	RF.9
Nombre requisito	Detección de conflictos entre los disparadores
Características	No todos los disparadores afectan unicamente a un trozo de texto bien delimitado.
Descripción requirimiento	El lenguaje natural no es sencillo de modelar y muchas veces se presentarán conflictos entre las diferentes etiquetas.
Prioridad	Alta

Tabla 3.9: Detección de conflctos

Id requisito	RF.10
Nombre requisito	Cálculo del alcance de los disparadores
Características	Etiquetar las oraciones sin añadir términos innecesarios y sin excluir ninguno.
Descripción requirimiento	En una misma oración se pueden presentar múltiples disparadores. Debemos ser capaces de separarlos y etiquetar solo las partes a las que afecten.
Prioridad	Alta

Tabla 3.10: Calculo de alcance

3.3. Requisitos no funcionales

Id requisito	RNF.1
Nombre requisito	Interoperabilidad
Características	Capacidad de acoplarse a código ya existente.
Descripción requirimiento	El algoritmo podrá formar parte de código ya desarrollado, por lo que una buena interoperabilidad es imprescindible.
Prioridad	Alta

Tabla 3.11: interoperabilidad

CAPÍTULO 4

SOLUCIÓN DEL PROBLEMA

4.1. Diseño

A continuación se presenta el diagrama de flujo de los diferentes procesos del algoritmo que hemos implementado. En la sección 4.2 se detalla la implementación de cada uno de los módulos.

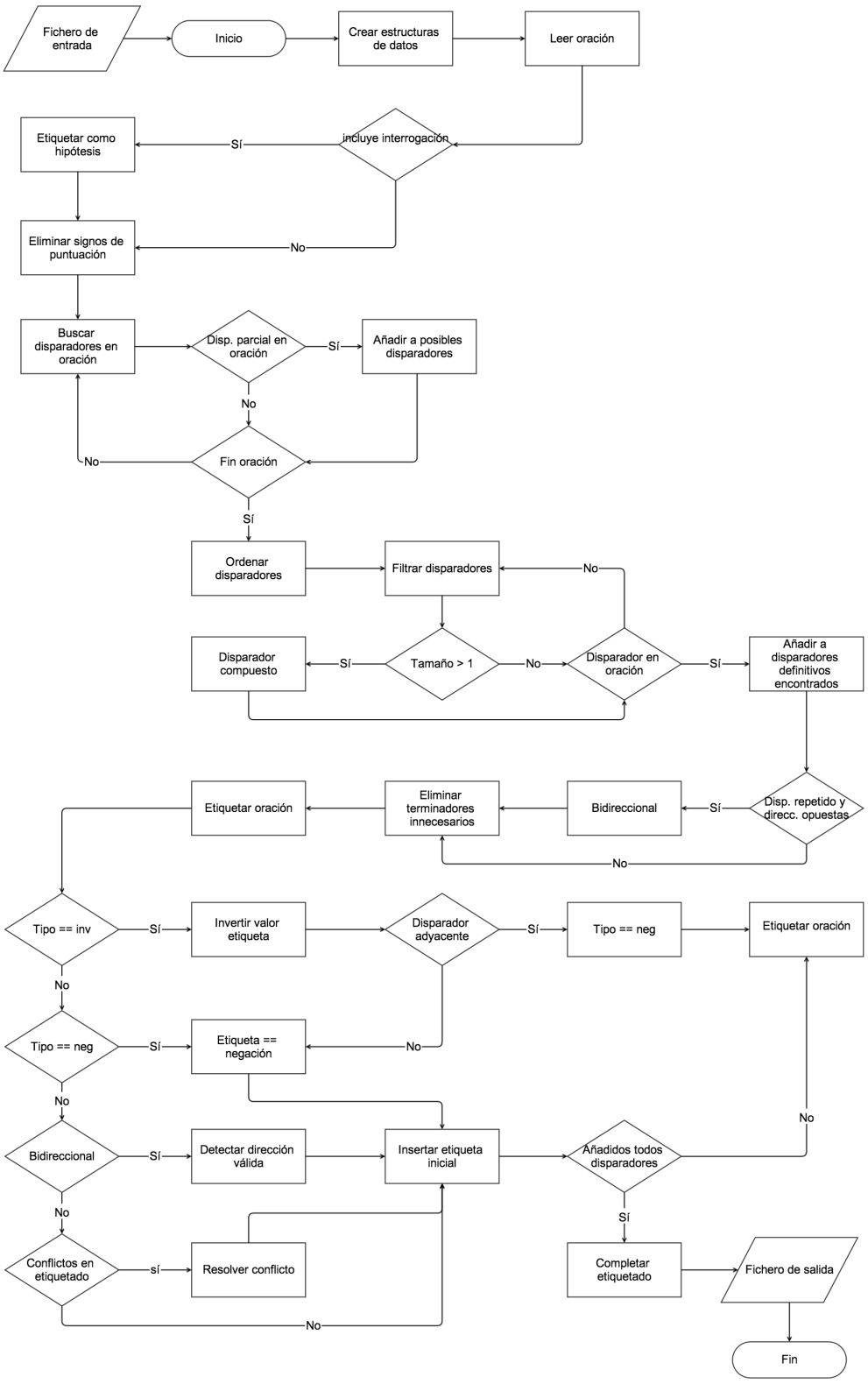

Figura 4.1: Diagrama de flujo

4.2. Implementación

En esta sección presentamos los detalles de la implementación del algoritmo y de cada uno de sus módulos.

4.2.1. Listado de posibles disparadores en notas médicas

Necesitamos ser capaces de determinar qué conjunto de palabras (disparadores) provocan que una oración, o parte de ella, sea etiquetada como negación, histórico, hipotético, etc. Para conseguirlo se analizaron de forma manual notas clínicas escritas en español extrayendo aquellas palabras que funcionaban como disparadores y se clasificaron por tipo (existencia definitiva, existencia probable, negación, histórico, etc.) y dirección (pre ó post). Actualmente se han detectado y clasificado 750 términos distintos que consideramos disparadores.

El algoritmo [1] recibe como entrada una nota clinica en la que las diferentes oraciones que la forman se encuentran en diferentes saltos de línea y el listado de los diferentes disparadores obtenido anteriormente. El resultado obtenido será otro fichero con las oraciones que componen la nota clínica con la diferente sintomatología médica etiquetada por su tipo. Una vez dispongamos de la lista de disparadores lo recorreremos en su totalidad para crear una estructura de datos [línea 3] (una lista de tripletas <String, String, String>) donde guardaremos el propio disparador, su tipo y dirección para posteriormente poder recorrerlo fácilmente.

Algoritmo 1: Crear estructuras de datos

Data: Fichero con las notas clínicas a etiquetar
Result: Fichero con las notas clínicas etiquetadas

```
1 Inicio;
2 forall the Lista de disparadores do
3 |   Crear estructura de datos con los disparadores, su tipo y su dirección;
4 end
```

4.2.2. Nomenclatura del etiquetado

Una de las primeras decisiones que hubo que tomar fue el tipo de nomenclatura a utilizar en el etiquetado de las notas médicas. Se decidió utilizar un sistema similar al lenguaje XML [Dom] , donde la apertura y cierre de las etiquetas se efectúa de la siguiente manera: <apertura>oración</cierre>, ya que nos permite mostrar de forma clara y concisa el tipo de etiquetado y su alcance.

4.2.3. Alcance del etiquetado

Para poder etiquetar correctamente las notas clínicas necesitábamos ser capaces de determinar el alcance de las etiquetas. Dicho alcance se determina bien porque se alcanza la siguiente palabra clave en la oración, bien porque la oración llega a su fin o bien porque encontramos un elemento terminador (por terminador entendemos aquellas disparadores que no inician un etiquetado nuevo pero finalizan el anterior ya que cambia el contexto de la oración).
Ej: "El paciente se mostraba consciente, orientado globalmente, con deshidratación cutáneo-mucosa y palidez cutánea, aunque con mucosas normocoloreadas". El término "aunque" no crea una etiqueta nueva pero cambia el contexto del etiquetado hasta el momento.

4.2.4. Detección de disparadores en notas médicas

Antes de proceder a la anotación de los textos clínicos necesitamos disponer de un listado de todos aquellos términos que pueden ser susceptibles de iniciar el etiquetado en una oración. Una vez obtenida dicha lista necesitamos ser capaces de encontrar a los candidatos; esta búsqueda se realiza comparando cada elemento de la lista de candidatos con cada una de las palabras que componen la oración extraída de la nota clínica, teniendo en cuenta que si el número de palabras no coincide o son palabras distintas automáticamente avanza al siguiente. Además un mismo fichero puede incluir múltiples notas clínicas separadas por párrafos . Debemos ser capaces de detectarlas y devolverlas anotadas en diferentes ficheros si el usuario así lo desea.

En el algoritmo [2] vemos que por cada oración del párrafo debemos recorrer cada una de las palabras para determinar si se trata de un disparador o no. Antes de comprobarlo primero verificaremos si aparece algún signo de interrogación [línea 6], ya que significaría que el contenido que queremos etiquetar es una pregunta, y por tanto de tipo *hipótesis*. Si no se trata de una oración interrogativa procederemos a eliminar signos de puntuación y convertiremos todas las letras mayúsculas a minúsculas [línea 9]. Esto es necesario porque en el listado de disparadores todas las palabras se encuentran guardadas en minúsculas y sin signos de puntuación. Por cada correlación que encontremos almacenaremos el posible disparador y una vez recorridos todos los disparadores ordenaremos las posibles coincidencias de mayor a menor número de palabras para su posterior comprobación con el contenido de la oración a etiquetar [línea 15].

Algoritmo 2: Detección de posibles disparadores

```
1  forall the Párrafos del documento do
2      Leer primer párrafo;
3      forall the Líneas del párrafo do
4          forall the Palabras de la oración do
5              if Palabra incluye signos de interrogación then
6                  Etiqueta = hipótesis;
7              end
8          end
9          Eliminar espacios, signos de puntuación y letras mayúsculas;
10         forall the Disparadores do
11             if Disparador incluye palabra then
12                 Añadir a lista de posibles disparadores;
13             end
14         end
15         Ordenar posibles disparadores por número de palabras, de mayor a menor;
16     end
17 end
```

4.2.5. Filtrar disparadores candidatos

Una vez detectados todos los posibles disparadores debemos comprobar si alguno de ellos coincide con la expresión incluida en la oración que estamos etiquetando. La búsqueda y comprobación de los disparadores se realiza en el algoritmo [3] en dos fases distintas por la presencia de disparadores compuestos (de más de una palabra) que además pueden incluir disparadores sencillos (de una única palabra).
Ej: “Con antecedentes de ” incluye el disparador ‘con’ pero sus significados son histórico y afirmado respectivamente.

Si el posible disparador es de longitud mayor a uno [línea 3] deberemos comprobar si las siguientes palabras de la oración coinciden. En caso afirmativo habremos detectado un disparador compuesto y asignaremos la etiqueta correspondiente. En caso contrario pasaremos al siguiente disparador. Si el disparador es longitud uno [línea 9] comprobaremos si son iguales, y procederemos de forma análoga al disparador compuesto.

Si encontramos la misma expresión en la oración más de una vez en la lista de posibles disparadores, puede tratarse de un disparador con varios sentidos o con varios significados[1]. En el caso de que el significado sea el mismo en las coincidencias marcaremos el disparador como bidireccional [línea 12].

[1] Afirmado, negado, histórico, etc.

Algoritmo 3: Filtrar disparadores candidatos

```
1  forall the Posibles disparadores do
2      if longitud >1 then
3          Comparar posibles disparadores compuestos;
4          if palabra == disparadorCompuesto then
5              Asignar etiquetas a la palabra según su tipo y posición en la oración;
6          end
7      end
8      if palabra == disparador then
9          Asignar etiquetas a la palabra según su tipo y posición en la oración;
10     end
11     if Disparador encontrado más de una vez AND significado igual then
12         Disparador bidireccional = true;
13     end
14 end
```

4.2.6. Eliminar terminadores no necesarios

En [4] comprobamos si el primer disparador es de tipo 'terminal' y el siguiente es de tipo 'pegajoso' [línea 2]. En ese caso el disparador 'terminal' no es necesario por lo que lo eliminaremos para que no afecte al etiquetado de la oración que estamos analizando.

Algoritmo 4: Eliminar terminadores no necesarios

```
1  if primer disparador == terminate AND siguiente disparador == sticky then
2      Eliminar disparador;
3  end
```

4.2.7. Sentido del etiquetado e incompatiblidades entre las mismas

El alcance de las disparadores puede ser pre o post. Hay que tener en cuenta esta característica para determinar si diferentes etiquetados son compatibles entre sí ya que existe la posibilidad de que se solapen entre ellos, teniendo que elegir cual de los dos utilizar o simplemente ignorando uno de ellos. La prioridad del tipo de etiqueta hay que realizarlo teniendo en cuenta que la negación tiene prioridad sobre la afirmación.
Ej: "El paciente muestra síntomas de fiebre ausente". "Síntomas" etiqueta los elementos a su derecha de forma positiva. Sin embargo "ausente" es un término de negación que afecta a los elementos a su izquierda. Los términos negados tienen prioridad sobre los afirmados por lo que el etiquetado final es "El paciente muestra <negado> síntomas de fiebre </negado> ausente."

En el algoritmo [5] comprobamos si el disparador actual es de dirección 'pre' y el siguiente disparador es de dirección 'post'. En el caso de que el siguiente disparador sea de tipo negado y el actual sea de tipo afirmado, puede existir un conflicto entre las etiquetas, teniendo prioridad el etiquetado negativo [línea 3]. Si el siguiente tipo de disparador igual al actual no será necesario dado que son redundantes [línea 7].

Algoritmo 5: Conflictos en etiquetado

```
1  if NOT bidireccional then
2      if siguiente disparador == post then
3          if siguiente disparador == negado then
4              posible conflicto entre etiquetas;
5          else
6              if siguiente disparador == disparador then
7                  etiqueta redundante;
8              end
9          end
10     else
11         etiquetar oracion;
12     end
13 end
```

4.2.8. Etiquetado múltiple

En lenguaje natural una palabra puede tener diferentes significados según el contexto en la que se utilice. Esto provoca que no todas las disparadores tengan un único significado. Para solucionar esta situación recurrimos al etiquetado anidado: un término médico puede estar rodeado de varias etiquetas dispuestas de forma consecutiva siguiendo la nomenclatura de balanceado de paréntesis (las etiquetas se añaden de dentro hacia fuera).
Ej: En la oración "Paciente de 41 años con antecedentes de tabaquismo severo.", el término "con antecendentes de" puede adquirir un significado referido a un síntoma pasado reflejado en el historial que es posible que continúe en la actualidad. Por tanto el contenido referido por el término "con" es necesario etiquetarlo como <afirmado><historico>dificultad respiratoria aguda</historico></afirmado>ya que no podemos descartar ninguna de las dos posibilidades.

En el algoritmo [2] detectamos los posibles disparadores. Si entre los candidatos encontramos dos disparadores iguales, con la misma dirección pero diferente tipo, querrá decir que ese disparador tendrá asociado varias etiquetas (algoritmo[6]).

Algoritmo 6: Etiquetado múltiple

```
1 forall the disparadores detectados do
2     if disparadores iguales AND direccion iguales then
3         disparador múltiple;
4     end
5 end
```

4.2.9. Disparadores bidireccionales

Al igual que en el caso del etiquetado múltiple, existen términos que pueden etiquetar tanto como pre como post. En estos casos se analizará si el término permite etiquetar en sentido pre. En caso contrario se analizará el posible etiquetado en sentido post.
Ej: "Persistió el dolor en el paciente" y "Inflamación y fiebre persistió tras el tratamiento". En ambos casos el término "persistió" está presente, pero el sentido es distinto en cada una de las oraciones.

El caso bidireccional (algoritmo [7]) es uno de los más complicados de tratar dada la gran cantidad de casos a tener en cuenta entre el propio disparador, el contexto en el que se sitúa y sus posibles confilctos. Para detectar si es bidireccional comprobaremos el siguiente disparador. Por la forma en la que almacenamos los disparadores (<etiqueta><negado><pre>, <etiqueta><negado><pos>), si es el mismo significará que es bidireccional y procederemos a averiguar en qué sentido debemos etiquetar [línea 1].

El primer paso será determinar en cuál de los dos disparadores repetidos nos encontramos para saber cuántas posiciones de la lista avanzar para extraer el siguiente disparador [línea 4]. Una vez calculado, comprobaremos que su sentido no sea 'post' y trataremos de etiquetar hacia la derecha. Si lo conseguimos habremos terminado y podremos eliminar el disparador repetido ya que no es necesario[línea 7]. Si no hemos podido etiquetar realizaremos un proceso similar al anterior (comprobar si el anterior es igual al disparador en el que nos encontramos y comprobar si su dirección no es 'pre' [línea 14]). En este caso etiquetaremos hacia la izquierda y eliminaremos el siguiente disparador. Si la dirección sí es 'pre' trataremos de etiquetar hacia la derecha en el caso de ser ese el sentido de la etiqueta actual, y eliminaremos el siguiente disparador por no seguir siendo necesario [línea 25]. Si no hemos conseguido etiquetar la oración de ninguna de las formas previas eliminaremos el disparador actual y probaremos con el siguiente en la próxima iteración [línea 29].

Algoritmo 7: Disparador bidireccional

```
1  if disparador == siguiente disparador then
2  |  bidireccional = true;
3  end
4  if disparador == siguiente disparador then
5  |  offset = 2;
6  end
7  if disparador + offset NOT post then
8  |  etiquetada = etiquetar oracion(pre);
9  |  if etiquetada then
10 |  |  eliminar siguiente disparador;
11 |  end
12 end
13 if NOT etiquetada then
14 |  if disparador == anterior disparador then
15 |  |  offset = 2;
16 |  end
17 |  if disparador - offset NOT pre then
18 |  |  if disparador == post then
19 |  |  |  etiquetar oracion(post);
20 |  |  |  eliminar siguiente disparador;
21 |  |  end
22 |  else
23 |  |  if disparador == pre then
24 |  |  |  etiquetar oracion(pre);
25 |  |  |  eliminar siguiente disparador;
26 |  |  end
27 |  end
28 end
29 if NOT etiquetada then
30 |  eliminar disparador;
31 end
```

4.2.10. Disparadores inversores

Existen disparadores como “salvo”, que modifican el tipo del siguiente disparador en el caso de existir uno previo. Si detectamos uno procederemos según la implementación detallada en el algoritmo [8] para resolverlo correctamente.
Ej: “Síntomas de enfermedad presente en el paciente salvo inflamación”. Enfermedad

deberá ser etiquetado como afirmado e inflamación como negado.

Algoritmo 8: Disparadores inversores

```
1 if Etiqueta == inv then
2     if existe disparador previo then
3         if disparador positivo then
4             disparador == negativo;
5         else
6             disparador == positivo;
7         end
8     end
9 end
```

4.2.11. Signo de interrogación

La inclusión de interrogaciones en una oración modifica completamente su significado. Es por ello que si detectamos la apertura de una interrogación todo el texto incluido hasta su cierre será etiquetado como hipótesis. El algoritmo presupone que la inserción de los signos de interrogación ha sido correcta y que existe tanto el signo de apertura como el signo de cierre de interrogación. La implementación se detalla en el algoritmo [2].

4.2.12. Negación de disparadores

Se ha optado por implementar una negación inteligente que sea capaz de detectar si un disparador de negación afecta a la siguiente palabra clave o no, y modificar el tipo de etiqueta por una negativa en el caso de que sea necesario.
Ej: “Los pulsos centrales y periféricos son simétricos y no hay edemas”. ‘No’ modifica el tipo de etiquetado de ‘hay’, convirtiendolo de una etiqueta afirmada a una negada. Hay que tener en cuenta que el disparador de negación no necesariamente afecta al siguiente disparador sino que puede negar los términos clínicos a los que precede.

En el algoritmo [9], si detectamos un disparador que provoca una negación (no, sin, en ningún momento, etc.), y tiene otro disparador adyacente [línea 3], se guardará esta situación y se tendrá en cuenta para la siguiente iteración del bucle. En el caso de no existir un disparador adyacente, la negación se aplicará al fragmento de la sentencia que corresponda a dicho disparador [línea 5].

Algoritmo 9: Disparadores de negación

```
1 if Etiqueta == neg then
2     if Existe disparador adyacente then
3         negar siguiente disparador == True;
4     else
5         negar fragmento de la oración correspondiente a la dirección del disparador
          actual;
6     end
7 end
```

4.2.13. Doble negación

En el algoritmo [10] comprobamos si en la iteración anterior se ha determinado que el siguiente disparador tiene que modificar su significado por una negación [línea 1]. De ser así, el disparador se etiquetará como 'negación', salvo si su significado previo ya era negación y no posibilita la negación de otros disparadores (en ese caso su significado pasará a ser 'probable') [línea 3].
Ej: En la oración "Los hallazgos radiológicos sugerían la presencia de meningioma aunque no se puede descartar otras posibilidades como un tumor de la vaina nerviosa", 'se puede descartar' es un disparador de negación, pero al estar en presencia de otro disparador de negación su significado se modifica a 'probable'. Sin embargo, en la oración "Mujer de 62 años sin ningún signo de herpes zoster" existe una doble negación pero su significado sigue siendo negado. Los disparadores 'sin' y 'ningún' niegan por si solos pero tambíen posibilitan la opción de que nieguen a otros disparadores por lo que su significado no cambia.

Algoritmo 10: Doble negación

```
1 if Negar siguiente disparador then
2     if disparador == negado then
3         disparador == probable;
4     else
5         disparador == negado;
6     end
7 end
```

4.2.14. Disparadores "pegajosos"

En lenguaje natural ciertos términos como 'y' u 'o' siguen un comportamiento distinto al resto. En el caso de 'y' extiende la etiqueta de su palabra clave predecesora (Pacien-

te diagnosticado de <afirmado>dolor</afirmado> y <afirmado>fiebre</afirmado>). Para "o" su comportamiento varía según la clase de etiqueta de su predecesor: si es de tipo negado 'o' extiende el tipo de etiqueta de su predecesor. En caso contrario actúa como un indicador de posibilidad, etiquetando como <probable></probable> el contenido a su derecha.
A continuación se muestran dos ejemplos: "No se constatan otras tumoraciones, ascitis, ni signos de insuficiencia hepática o hipertensión portal" nos permite observar que el término 'o' extiende la negación de los términos anteriores. Sin embargo en la oración "La biopsia informaba de células con inclusiones por cuerpo extraño sugestivas de enfermedad maligna o de dermatofitosis" el término 'o' define varias posibilidades y el término médico adyacente a su derecha ha de ser etiquetado como probable.

Para los disparadores pegajosos (algoritmo [11]) deberemos buscar el tipo de etiqueta del disparador anterior en caso de existir y extender su alcance [línea 2]. Si no existe y el disparador pegajoso es 'y', se etiquetará el contenido situado a su izquierda como afirmado [línea 12]. Si se trata de otro disparador se etiquetará como probable [línea 8].

Algoritmo 11: Disparadores "pegajosos"

```
1  for i = tagIntex, i > 0, i − − do
2      buscar disparador anterior;
3      if encontrado then
4          extender disparador;
5          if disparador == 'y' then
6              etiqueta == afirmado;
7          else
8              etiqueta == probable;
9          end
10     else
11         if disparador == 'y' then
12             etiquetar oración;
13         end
14     end
15 end
```

4.2.15. Inserción de etiquetas

Tras determinar el alcance, sentido y tipo de etiquetado procedemos a insertarlo en la oración. Nuestro algoritmo buscará las disparadores cuya existencia ya ha sido determinada en la oración y procederá a añadir la etiqueta correspondiente a la izquierda/cierre o derecha/apertura de dicha palabra según corresponda su sentido (pre/post).

A la hora de insertar las etiquetas en la oración [12] procederemos de la siguiente forma: recorremos la oración a insertar hasta encontrar el primer disparador [línea 2]. Una

vez seleccionado comprobaremos si se trata de un disparador "pegajoso" (extienden el tipo de etiqueta del disparador anterior). En caso afirmativo procederemos a determinar qué etiqueta tenemos que extender [línea 4] como se detalla en el algoritmo [11]. Si no es así comprobaremos la dirección de la etiqueta y la añadiremos en consecuencia, no sin antes comprobar que la palabra anterior (en caso de ser post) o posterior (en caso de ser pre), no sea ya una etiqueta o un disparador [línea 6]. De ser así no habría que insertar ninguna etiqueta para el disparador que estemos analizando.

Algoritmo 12: Insertar etiquetas

```
1  forall the Palabras de la oración do
2      buscar etiqueta;
3      if pegajoso then
4          buscar disparador previo;
5      end
6      if disparador == post then
7          if (anterior NOT etiqueta) AND (anterior NOT disparador) then
8              etiquetar oracion;
9          end
10     else
11         if (siguiente NOT etiqueta) AND (siguiente NOT disparador) then
12             etiquetar oracion;
13         end
14     end
15 end
```

4.2.16. Completar etiquetado

Una vez añadidas las etiquetas de apertura o cierre en la fase anterior, procederemos a incluir las etiquetas restantes para cerrarlas (algoritmo [13]). Para ello recorreremos todos los disparadores detectados previamente y, empezando desde el primero, buscaremos su posición en la oración a etiquetar. Una vez encontrado insertaremos la etiqueta de cierre [línea 3] siguiendo el mismo procedimiento que en [12].

Algoritmo 13: Completar etiquetado

```
1  forall the Disparadores encontrados do
2      if disparador encontrado en oración then
3          etiquetar oración;
4      end
5  end
```

Finalmente escribiremos los resultados obtenidos en un fichero de texto e imprimire-

mos por pantalla las oraciones anotadas.

4.3. Algoritmo propuesto

Los diferentes procedimientos se invocarán en el orden especificado a continuación hasta etiquetar todas las notas clínicas facilitadas como entrada del programa.

Algoritmo 14: Programa principal

Data: Fichero con las notas clínicas a etiquetar
Result: Fichero con las notas clínicas etiquetadas

```
1  Inicio;
2  forall the Oraciones del fichero con las notas clínicas a etiquetar do
3      procedure EstructuraDatos(ListadoDisparadores);
4      procedure DetectarPosiblesDisparadores(Oración, EstructuraDisparadores);
5      procedure FiltrarDisparadores(Oración, PosiblesDisparadores);
6      procedure EliminarTerminadoresInnecesarios(ListadoDisparadoresFinal);
7      procedure DisparadorInversor(Oración, ListadoDispadoresFinal);
8      procedure DisparadorPegajoso(Oración, ListadoDispadoresFinal);
9      procedure DisparadorNegación(Oración, ListadoDispadoresFinal);
10     procedure DobleNegación(Oración, ListadoDispadoresFinal);
11     procedure DisparadorBidireccional(Oración, ListadoDispadoresFinal);
12     procedure EtiquetadoNormal(Oración, ListadoDispadoresFinal);
13     procedure ConflictoEtiquetado(Oración, ListadoDispadoresFinal);
14     procedure InsertarEtiquetas(Oración, ListadoDispadoresFinal);
15     procedure CompletarEtiquetado(Oración, ListadoDispadoresFinal);
16 end
```

4.4. Detalles de la implementación

El lenguaje de programación que hemos utilizado es Java dada su popularidad y la gran cantidad de programas ya escritos en este lenguaje en el ámbito de la biotecnología. Esto facilita enormemente su cohesión con módulos ya existentes o aquellos que se puedan implementar en un futuro.

Hemos recurrido a la librería externa Javatuples [Fer11] ya que nos proporciona la capacidad de crear y manipular fácilmente tuplas de tamaño N, las cuales utilizamos para almacenar los disparadores que hemos encontrado en la oración a etiquetar junto a información necesaria como su tipo, dirección y posición relativa en la oración o la posición de los diferentes disparadores en la oración ya que un disparador repetido puede significar que sea bidireccional o simplemente que aparezca de forma repetida en la oración.

CAPÍTULO 5

VALIDACIÓN

5.1. Validación

Para la validación del algoritmo se han usado textos clinicos extraidos de la base de datos SciELO [Sci]. En concreto se han extraido 454 oraciones de textos clínicos. Estas oraciones contienen concretamente un total de 1897 disparadores divididos en 1507 afirmaciones, 227 negaciones, 107 probables, 2 negaciones probables y 124 históricos.

Para conseguir una validación no sesgada primero se etiquetaron de forma manual todas las oraciones, buscando "hechos" relevantes en ambientes médicos y anotándolos para, según su entorno y su contexto en la oración, si son afirmados, negados, probables, etc. Posteriormente se aplicó el algoritmo a las mismas oraciones y se compararon una a una la salida obtenida y la salida esperada por cada una de ellas. En el caso de ser la misma se marcaba como 'acierto', y si eran distintas se marcaban como 'error'. Todos los disparadores que han etiquetado síntomas no clínicos, como artículos o fechas, también se han incluido como errores de etiquetado. Los resultados obtenidos son:

Número oraciones	**Número términos**	**Aciertos**	**Porcentaje aciertos**	**Fallos**	**Porcentaje fallos**
429	1897	1589	78.4 %	437	21.6 %

Tabla 5.1: Número total de aciertos

Términos médicos	**Número de términos**	**Aciertos**
Afirmados	1507	1294
Negados	227	240
Probables	107	118
Probable negado	2	6
Históricos	124	99

Tabla 5.2: Aciertos por tipo de etiqueta

Para poder calcular la precisión, sensibilidad, valor-f y exactitud del algoritmo con los resultados obtenidos primero debemos calcular los TP (acierto positivo), TN (negativo positivo), FP (falso positivo) y FN (falso negativo).

- **TP:** términos etiquetados correctamente.
- **TN:** términos no etiquetados correctamente.
- **FP:** términos etiquetados por el algoritmo de forma errónea.
- **FN:** términos que no han sido etiquetados por el algoritmo a pesar de ser necesario.

Términos médicos	TP	FP	TN	FN
Afirmados	1244	50	345	263
Negados	201	39	1388	26
Probables	87	31	1502	20
Probable negado	2	4	1587	0
Históricos	55	44	1537	69

Tabla 5.3: Aciertos y fallos positivos y negativos

Una vez obtenidos los valores podremos calcular la precisión, sensibilidad, valor-f y exactitud del algoritmo utilizando las fórmulas matemáticas especificadas a continuación. Los valores obtenidos se pueden consultar en [5.4].

$$Precision = \frac{TP}{TP + FP}$$

$$Sensibilidad = \frac{TP}{TP + FN}$$

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Valor - F = \frac{2 * (Precision * Exactitud)}{Precision + Exactitud}$$

Términos médicos	Precisión	Sensibilidad	Valor F	Exactitud
Afirmados	96.1 %	82.5 %	88.8 %	83.5 %
Negados	83.8 %	88.5 %	86.1 %	96.1 %
Probables	73.7 %	81.3 %	77.3 %	96.9 %
Probable negado	33.3 %	100 %	50 %	99.7 %
Históricos	55.6 %	44.4 %	49.3 %	93.4 %

Tabla 5.4: Precisión, sensibilidad, valor-f y exactitud

Comparando la precisión (P), valor-f (F) y la exactitud (E) del algoritmo ConText en textos clínicos escritos en inglés con nuestra implementación podemos observar un rendimiento peor (sobre todo en la detección de términos históricos), pero que consideramos suficiente para una primera aproximación.

	Neg.			**Prob.**			**Hist.**		
	P	F	E	P	F	E	P	F	E
Español	96 %	86 %	96 %	73 %	76 %	96 %	55 %	49 %	93 %
Inglés	96 %	95 %	93 %	93 %	76 %	65 %	86 %	82 %	86 %

Tabla 5.5: Comparación resultados español e inglés

CAPÍTULO 6

CONCLUSIONES

6.1. Conclusiones

En este TFG se ha presentado un algoritmo que permite, en textos clínicos escritos en español:

1. Detectar términos clínicos afirmados.
2. Detectar términos clínicos negados.
3. Detectar términos clínicos probables.
4. Detectar términos clínicos referidos al historial del paciente.

El algoritmo presentado se ha implementado en JAVA y se ha probado con 454 oraciones y 1897 disparadores obteniendo 1589 aciertos (78.4 %) y unos valores de precisión, sensibilidad, valor-f y exactitud del 96.1 %, 82.5 %, 88.8 % y 83.5 % respectivamente para términos médicos afirmados; un 83.8 %, 88.5 %, 86.1 % y 96.1 % para términos negados; un 73.7 %, 81.3 %, 77.3 % y 96.9 % para términos probables; 33.3 %, 100 %, 50 % y 99.7 % para términos probables negados y un 55.6 %, 44.4 %, 49.3 % y 93.4 % para términos históricos.

Consiguientemente los objetivos planeados al comienzo de estes TFG se han conseguido con exito.

6.2. Líneas futuras

A pesar de que los objetivos se han superado con éxito, la realización de este TFG deja abiertas lineas de trabajo futuro relacionadas principalmente con la mejora del rendimiento del algoritmo. En particular las siguientes líneas que se plantean contribuirían a mejorar el rendimiento del algoritmo presentado:

- **Mejorar la lista de disparadores.** Una lista de disparadores más amplia mejoraría los resultados de la búsqueda de disparadores en las oraciones.
- **Mejoras en la detección de los múltiples formatos de fechas.** Dada la gran cantidad de formatos en los que se puede escribir una fecha, ser capaces de detectaras mejoraría la detección de términos clínicos de tipo histórico.
- **Cambios de contexto mediante signos de puntuación.** El cambio de contexto en una oración puede producirse por una simple coma. Ser capaces de detectar dichos cambios mejoraría el rendimiento del algoritmo.

Bibliografía

[CBH+01] Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*, 2001:34–301, 2001.

[CBH+02] Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. Evaluation of negation phrases in narrative clinical reports, 2002.

[Cer13] Insituto Cervantes. El español: una lengua viva. `http://eldiae.es/wp-content/uploads/2013/06/2013_espanol_lengua_viva.pdf`, 2013.

[CHV+13] Wendy Webber Chapman, Dieter Hillert, Sumithra Velupillai, Maria Kvist, Maria Skeppstedt, Brian E. Chapman, Mike Conway, Melissa Tharp, Danielle Mowery, and Louise Deleger. Extending the negex lexicon for multiple languages. volume 192 of *Studies in Health Technology and Informatics*, pages 677–681. IOS Press, 2013.

[CW07] Dowling JN Chapman WW, Chu D. Context: An algorithm for identifying contextual features from clinical text. *BioNLP workshop of the association for computational linguistics*, 2007.

[Dom] Ubiquitous Web Domain. Extensible markup language.

[EBB+05] Peter L. Elkin, Steven H. Brown, Brent A. Bauer, Casey S. Husser, William Carruth, Larry Bergstrom, and Dietlind Wahner-Roedler. A controlled trial of automated classification of negation from clinical notes. *BMC Med. Inf. andDecision Making*, 5, 2005.

[FC99] Hripcsak G. Friedman C. Natural language processing and its future in medicine. *Acad Med*, 1999.

[Fer11] Daniel Fernández. Javatuples. `http://www.javatuples.org/`, 2011.

[HDTC09] Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy Webber Chapman. Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839–851, 2009.

[HL07] Yang Huang and Henry J. Lowe. A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American Medical Informatics Association*, 14(3):304–311, 2007.

[MD09] Roser Morante and Walter Daelemans. A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 21–29. Association for Computational Linguistics, 2009.

[MDN01] Pradeep Mutalik, Aniruddha M. Deshpande, and Prakash M. Nadkarni. Research paper: Use of general-purpose negation detection to augment concept indexing of medical documents: A quantitative study using the umls. *JAMIA*, 8(6):598–609, 2001.

[MdS13] Servicios Sociales e Igualdad Ministerio de Sanidad. E-Health - Nota de prensa. `http://www.msssi.gob.es/gabinete/notasPrensa.do?id=2968`, 2013.

[MLD08] Roser Morante, Anthony Liekens, and Walter Daelemans. Learning the scope of negation in biomedical texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 715–724, 2008.

[MS14] Gunnar H. Nilsson Hercules Dalianis Maria Skeppstedt, Maria Kvist. Automatic recognition of disorders, findings, pharmaceuticals and body structures from clinical text: An annotation and machine learning study. *Journal of Biomedical Informatics. In Press.*, 2014.

[NPCD12] Jacinto Mata Vázquez Victoria Pachón Álvarez Noa P. Cruz Díaz, Manuel J. Maña López. A machine-learning approach to negation and speculation detection in clinical texts. *Journal of the American Society for Information Science and Technology*, 2012.

[Org] International Health Terminology Standards Development Organisation.

[RRM08] Lior Rokach, Roni Romano, and Oded Maimon. Negation recognition in medical narrative reports. *Inf. Retr.*, 11(6):499–538, December 2008.

[Sci] SciELO. Scientific electronic library online.

[SDN11] Maria Skeppstedt, Hercules Dalianis, and Gunnar H. Nilsson. Retrieving disorders and findings: Results using snomed ct and negex adapted for swedish. In *LOUHI 2011 Health Document Text Mining and Information Analysis 2011 : Proceedings of LOUHI 2011 Third International Workshop on Health Document Text Mining and Information Analysis Bled, Slovenia, July 6, 2011.*, number 744, pages 11–17, 2011.

[Ske11] Maria Skeppstedt. Negation detection in swedish clinical text: An adaption of negex to swedish. *J. Biomedical Semantics*, 2(S-3):S3, 2011.

[Sym] Symtex. `http://www.symtext.com/`.

[WWC13] Velupillai Kvist Skeppstedt Brian E. Chapman Conway Tharp Mowery Deleger Wendy W. Chapman, Hilert. Negex. https://code.google.com/p/negex/, 2013.

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Fri Jun 06 20:44:46 CEST 2014
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)