Extracción de Eventos y Expresiones Temporales en Textos Clínicos

Trabajo Fin de Máster

Juan Manuel Vicente Cabero

Trabajo de investigación para el

Máster Universitario en Tecnologías del Lenguaje

Universidad Nacional de Educación a Distancia

Dirigido por los profesores

Dra. Dª. Lourdes Araujo Serna

Dr. D. Juan Martínez Romo

Junio 2021

Agradecimientos

Quiero agradecer a mis tutores, Lourdes Araujo Serna y Juan Martínez Romo, por la orientación y el apoyo que me han proporcionado durante la realización del máster, y en especial en este trabajo.

También quiero agradecer a mi familia y amigos su apoyo incondicional, que me permite seguir creciendo a nivel personal y profesional.

Resumen

En el ámbito clínico, la identificación de eventos como la sintomatología de un paciente, las enfermedades que pueda padecer o los tratamientos que se le han aplicado, es una necesidad común entre los profesionales sanitarios e investigadores. Más allá de la detección de los eventos, es necesario poder situarlos en una línea temporal, pudiendo saber con facilidad el historial clínico del paciente, la sintomatología que presenta y el tiempo que ha pasado desde que se le administró la última dosis de un determinado fármaco.

La extracción automática del historial del paciente dado un informe sobre su ingreso en el hospital es un problema englobado dentro de la línea de investigación de la Extracción de Información, concretamente por la tarea de Extracción de Relaciones Temporales. Su aplicación al dominio clínico ha recibido atención especialmente en la última década en varias ediciones de congresos como i2b2 y Clinical TempEval, en las que se han presentado corpus y metodologías de evaluación para incentivar el desarrollo de la investigación del campo. Esta tarea tiene como antecedente la Extracción de Eventos y Expresiones Temporales, dado que son los elementos necesarios para extraer las relaciones.

En el presente documento se describe la tarea de Extracción de Eventos y Expresiones Temporales en Textos Clínicos, evaluando 5 arquitecturas que puedan abordar el problema partiendo de informes de alta de pacientes del corpus i2b2. Las arquitecturas toman como referencia las consideradas como estado del arte, tomando una de ellas como base y aplicando modificaciones de forma progresiva a fin de conseguir igualar o mejorar los resultados, a la par que se reducen el procesado y la cantidad de datos necesaria para entrenar dichos sistemas. Estas modificaciones dan lugar al sistema BertSR, que supone un planteamiento nuevo apoyado en un modelo BERT pre-entrenado, y que consigue igualar, e incluso superar en algunas subtareas, los resultados de los mejores sistemas del corpus i2b2.

Abstract

The identification of events in the clinical domain, such as a patient's symptomatology, the diseases he/she may suffer from or the treatments he/she has undergone, is a common need among healthcare professionals and researchers. Beyond the detection of events, it is necessary to be able to place them in a timeline, being able to easily know the patient's clinical record, the symptoms presented and the time that has passed since the last dose of a given drug was applied.

The automatic extraction of the patient's record given a medical admission report is a problem encompassed within the Information Extraction research line, specifically by the Temporal Relations Extraction task. Its application to the clinical domain has received attention especially in the last decade in several editions of conferences such as i2b2 and Clinical TempEval, where corpora and evaluation methodologies have been presented to encourage the development of research in the field. This task is based on Event and Time Expression Extraction, since these are the necessary elements to correctly retrieve the relationships.

This paper describes the task of Event and Temporal Expression Extraction in Clinical Texts, evaluating 5 architectures that can address the problem based on patient discharge reports from the i2b2 corpus. The architectures take as a reference those considered as state of the art, taking one of them as a base and applying modifications progressively in order to achieve equal or better results, while reducing the processing and the amount of data needed to train these systems. These modifications give rise to the BertSR system, which is a new approach to the problem, based on a pre-trained BERT model, and which manages to match, and even surpass in some subtasks, the results of the best systems in the i2b2 corpus.

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Motivación

En el ámbito clínico, la identificación de eventos como la sintomatología de un paciente, las enfermedades que pueda padecer o los tratamientos que se le están aplicando es una necesidad común entre los profesionales sanitarios e investigadores. Más allá de detectar únicamente estos eventos, es necesario poder situarlos en una línea temporal, para poder identificar el historial clínico del paciente, los posibles tratamientos que se hayan aplicado sin éxito, o cuánto tiempo ha pasado desde la última dosis que se le ha administrado de un determinado medicamento, a fin de evitar sobredosis o provocar combinaciones de medicamentos no deseadas en el organismo.

Estos documentos clínicos, como pueden ser los informes de alta, generalmente son redactados sin una estructura estandarizada, pudiendo mostrar variaciones en la redacción en función de la persona que los redacta, como el vocabulario, las abreviaturas o los formatos de fechas utilizados. Esto aumenta el tiempo que necesitan los profesionales para leer los documentos, y dificulta su procesamiento de forma automática.

Este problema se puede englobar dentro de la línea de investigación de Extracción de Información del campo del Procesado del Lenguaje Natural, que tiene como objetivo extraer información automáticamente de fuentes no estructuradas con el fin de presentarla de forma estructurada para facilitar un posterior procesado. Más concretamente, el problema de generar líneas temporales de eventos es abordado por la tarea de Extracción de Relaciones Temporales que, a su vez, requiere haber reconocido previamente todos los eventos y expresiones temporales del texto.

1.2. Propuesta y objetivos

A fin de solucionar el problema planteado anteriormente, en el presente documento se estudia la tarea de Extracción de Eventos y Expresiones Temporales en Textos Clínicos, examinando el estado del arte y proponiendo nuevos sistemas que permitan resolver la tarea de forma automática.

Los sistemas propuestos deben tener la capacidad de resolver las siguientes subtareas:

- ***Event Spans* (ES):** Detección de intervalos de texto (*spans*) que contienen eventos.
- ***Event Attributes* (EA):** Clasificación de todos los atributos que componen los eventos previamente detectados.
- ***Time Spans* (TS):** Detección de intervalos de texto que contienen expresiones temporales.
- ***Time Attributes* (TA):** Clasificación de todos los atributos que componen las expresiones temporales previamente detectadas.

En este trabajo se proponen un total de 5 sistemas para abordar las anteriores subtareas, todos ellos basados en aprendizaje supervisado, llegando a obtenerse resultados cercanos a los mejores sistemas propuestos hasta la fecha sobre el mismo corpus. Destaca especialmente el sistema basado en un modelo BERT pre-entrenado, sobre el que se aplica *fine-tuning* con dicho corpus para la detección de *spans*; en dos SVM para la clasificación de atributos y en un sistema basado en reglas para la normalización de fechas y horas. El sistema anterior consigue igualar a los mejores sistemas previamente presentados en algunas de las subtareas, e incluso consigue superarlos en otras.

1.3. Estructura del documento

El presente documento se estructura de la siguiente forma:

Capítulo 1. Introducción. Este capítulo introduce los principales motivos que han llevado a la realización de este trabajo, así como la propuesta y los objetivos del mismo.

Capítulo 2. Estado del arte. Este capítulo describe en mayor detalle la disciplina de la Extracción de Eventos y Expresiones Temporales en Textos Clínicos, presentando su origen y su evolución hasta el presente.

Capítulo 3. Sistemas propuestos. En este capítulo se describen en profundidad los sistemas propuestos para abordar el problema planteado, así como sus fundamentos.

Capítulo 4. Evaluación. Este capítulo describe la metodología utilizada para evaluar los diferentes sistemas propuestos, a la vez que presenta los resultados obtenidos al evaluarlos, y su comparación respecto a los mejores sistemas que utilizan el mismo corpus.

Capítulo 5. Conclusiones y trabajo futuro. Este capítulo recopila las diferentes conclusiones extraídas del trabajo realizado, y propone algunas líneas de trabajo futuro.

Capítulo 2

Estado del arte

Este capítulo describe en mayor detalle la disciplina de la extracción de eventos y expresiones temporales en textos clínicos, presentando su origen y su historia hasta el presente. Se muestran las técnicas actuales más utilizadas para resolver las tareas más relevantes del tema abordado, así como sus retos y líneas de trabajo en curso.

2.1. Descripción del problema

La tarea de Extracción de Eventos y Expresiones Temporales puede englobarse dentro del campo de la Extracción de Información (EI), consistente en la obtención de fragmentos de información relativa a un conjunto de conceptos relacionados entre sí, denominado escenario de extracción, sobre una serie de documentos (Turmo, 2004).

En la tarea de EI se recurre a fuentes de información no estructurada, como es el caso de los documentos de texto, con el objetivo de obtener información estructurada, definiéndose la información de interés y su formato de presentación en una plantilla de salida (Hobbs, 1993).

En el caso de la Extracción de Eventos y Expresiones Temporales en Textos Clínicos, y para la colección de datos utilizada, la plantilla se compondrá de dos tipos de entidades: `EVENTs` para los eventos y `TIMEX3` para las expresiones temporales. Los atributos de las entidades anteriores, así como la información que contienen, se describen en detalle junto con la colección de datos en la sección 2.3.

El reconocimiento de los eventos y las expresiones temporales suele servir como base para la extracción de las relaciones entre ambos tipos de entida-

des, permitiendo generar una línea temporal de los eventos recogidos en uno o varios documentos. No obstante, este trabajo se centra únicamente en la primera etapa, teniendo como objetivo la identificación de las entidades anteriormente mencionadas.

2.2. Trabajos previos

Los trabajos previos pueden dividirse en dos categorías en función del desarrollo del campo de estudio a lo largo del tiempo. En primer lugar, en la sección 2.2.1 se expone la tarea general de Extracción de Eventos y Expresiones Temporales, en el marco de las conferencias TempEval; y, posteriormente, se expone la adaptación de esta tarea al dominio clínico en la sección 2.2.2.

2.2.1. Extracción de Eventos y Expresiones Temporales

El progreso de una línea de investigación en muchos casos está ligado a la existencia de un conjunto de datos estándar que permita la evaluación de sistemas, de forma que estos sean comparables. El estado actual del campo de Extracción de Eventos y Expresiones Temporales tiene su origen en el esquema de anotación TimeML (Pustejovsky et al., 2003a), definido como candidato estándar de anotación para solucionar los cuatro problemas principales identificados para la extracción de eventos y expresiones temporales:

- La asociación de eventos a puntos concretos del tiempo.
- La ordenación de eventos.
- La capacidad de inferir el valor de expresiones temporales que no aparecen de forma explícita, como "la semana pasada" o "el día anterior".
- La capacidad de inferir la duración de un evento, o de sus consecuencias.

Para abordar los anteriores problemas, TimeML se apoya en cuatro estructuras de datos o entidades: `EVENT`, para los eventos, definidos como hechos que acontecen en un momento dado, o durante un periodo de tiempo; `TIMEX3`, para las expresiones temporales, que definen esas fechas, horas y duraciones, entre otros; `SIGNAL`, que permite anotar secciones del texto que

funcionan como marcadores, indicando de qué forma se relacionan las entidades entre sí, como podría ser el caso de "después" en la frase "María vendrá después de clase"; y `LINK`, que permite identificar los tres tipos de relaciones temporales existentes: `TLINK`, para relaciones temporales entre eventos, o eventos y expresiones temporales; `SLINK`, para las relaciones de subordinación entre eventos, o entre eventos y marcadores; y `ALINK`, para relaciones aspectuales (iniciación, culminación, terminación o continuación) entre eventos aspectuales y sus eventos argumentales.

A modo de prueba de concepto del esquema de anotación TimeML, en el mismo año se presentó el corpus TimeBank (Pustejovsky et al., 2003b), sirviendo como base para la detección de eventos, expresiones y relaciones temporales en varios proyectos durante los siguientes años (Mani et al., 2006; Boguraev et al., 2007).

TempEval

Además de la existencia de un estándar de anotación y de su aplicación en conjuntos de datos, es habitual recurrir a tareas y competiciones en congresos para incentivar el avance de las líneas de investigación en el campo del Procesado del Lenguaje Natural. En el caso de la Extracción de Eventos y Expresiones Temporales, en marco del congreso SemEval 2007 se plantean las bases para una tarea de evaluación, denominada TempEval, consistente en la identificación de eventos, expresiones y relaciones temporales contenidas en un texto (Verhagen et al., 2007).

En este primer ejercicio de evaluación se proponen 3 tareas (A, B y C) con potenciales aplicaciones, y realizables de forma automática. Para cada una de estas tareas se proporcionan conjuntos de datos de entrenamiento y evaluación con anotaciones, permitiendo identificar los límites de las oraciones, todas las expresiones temporales acordes al formato `TIMEX3`, todos los eventos acordes al formato definido en TimeML, y un conjunto de instancias de relaciones temporales relevantes para la tarea en cuestión. En las tareas A y B sólo se contemplaba un número limitado de eventos, aquellos que estuvieran presentes un mínimo de 20 ocasiones en el corpus TimeBank. El objetivo de todas las tareas era evaluar la extracción de relaciones temporales, siendo la extracción de eventos y expresiones temporales únicamente un paso previo, por lo que no se describen en profundidad al quedar fuera del dominio del presente trabajo. Puede encontrarse una descripción detallada

de las mismas en (Verhagen et al., 2007).

El esquema de anotación de la tarea TempEval se plantea como una versión simplificada de TimeML, utilizándose 5 posibles etiquetas: `TempEval`, `s`, `TIMEX3`, `EVENT` y `TLINK`. La etiqueta `TempEval` sirve para identificar la raíz del documento, y `s` para marcar el límite de las oraciones. La etiqueta `TIMEX3` es idéntica a la especificación planteada en TimeML, pero incluyendo un tipo especial de `TIMEX3`, denominado *Document Creation Time*, que se interpreta como un intervalo temporal que abarca al completo el día de creación del documento. La etiqueta `EVENT` también es idéntica a su definición de TimeML, permitiendo identificar los eventos del texto, generalmente denotados por verbos, sustantivos o adjetivos. Sus principales atributos son el tiempo verbal (`TENSE`), el aspecto del verbo (`ASPECT`), su modalidad (`MODALITY`) y la polaridad de la información que contiene (`POLARITY`). Por último, la etiqueta `TLINK` es una versión simplificada de su especificación en TimeML, definiendo sólo 6 tipos de relaciones temporales: `BEFORE`, para relaciones de precedencia; `AFTER`, para relaciones de sucesión; `OVERLAP`, para solapamientos; `BEFORE-OR-OVERLAP` y `OVERLAP-OR-AFTER`, para casos ambiguos de las anteriores; y `VAGUE` para los casos en los que no se puede determinar una relación específica. El conjunto de datos de anotaciones de `EVENTs` y `TIMEX3` utilizado en TempEval se obtiene del corpus TimeBank, aplicando las simplificaciones sobre su representación explicadas anteriormente.

TempEval-2

En el congreso SemEval 2010 volvió a incluirse una tarea en esta línea de investigación, partiendo de las bases definidas en TempEval, denominada TempEval-2 (Pustejovsky y Verhagen, 2009). Esta edición se diferencia de la anterior en que se plantea como una tarea multilingüe, proporcionando datos en inglés, español, italiano, chino y coreano; y que se compone de 6 subtareas en lugar de 3, con el objetivo de añadir una mejor cobertura al problema del procesado temporal de textos. Dado que el objetivo principal de esta tarea es el etiquetado de textos de forma que se obtenga una caracterización temporal lo más completa posible de los eventos que contiene, si el grafo de anotación de un documento no es completamente conexo no puede determinar las relaciones temporales entre dos eventos arbitrarios, dado que se pueden encontrar en diferentes componentes del grafo sin conexión entre

sí.

Este hecho motiva la ampliación de tareas, así como de sus descripciones, respecto a TempEval. Entre las 6 tareas definidas, dos de ellas resultan de especial interés por ser las precursoras de las tareas a resolver en el presente trabajo:

A. Detección de expresiones temporales acorde al formato `TIMEX3`, según su definición en TimeML. Se consideran expresiones temporales las que sintácticamente se muestren como locuciones adverbiales temporales, como "por la tarde", o preposicionales de tiempo, como "a las 10 de la mañana". Se detecta en primer lugar la extensión de las expresiones, es decir, el intervalo de caracteres que comprenden dentro del texto (*span*), así como sus atributos `TYPE` y `VAL`. El atributo `TYPE` determina el tipo de expresión, pudiendo tomar los valores `TIME`, `DATE`, `DURATION` y `SET`. El atributo `VAL` se corresponde con el valor normalizado de la expresión temporal, acorde al estándar ISO8601.

B. Detección de eventos, es decir, todos aquellos elementos de texto que puedan expresar eventualidad, generalmente introducidos por verbos y, en ocasiones, también por nombres, como "incendio" en la oración "ayer se declaró un incendio". Al igual que en el caso de las expresiones temporales, en primer lugar debe detectarse la extensión del evento en el texto, para detectar seguidamente sus atributos, tal y como se definen en TimeML: `TENSE`, `ASPECT`, `POLARITY` y `MODALITY`.

Las 4 tareas restantes (C-F) corresponden a la detección de relaciones temporales, por lo que quedan fuera del alcance de este documento, pudiendo consultarse sus descripciones en (Pustejovsky y Verhagen, 2009).

El conjunto de datos utilizado en TempEval-2 se basa en el estándar de anotación TimeML, en su versión 1.2.1. (Saurı et al., 2006), introduciendo marcadores de los eventos principales del texto como única diferencia respecto a TimeBank, el corpus utilizado en TempEval. Los datos pueden dividirse en dos subconjuntos, uno para las tareas A y B, y otro para el resto de tareas. Además, se desarrolla un corpus independiente para cada uno de los idiomas contemplados, sin tratarse de corpus paralelos.

TempEval-3

TempEval-3 (UzZaman et al., 2013) se plantea como continuación a las dos anteriores, cubriendo únicamente textos en inglés y en español. Presenta varias diferencias respecto a sus predecesoras, siendo destacables dos de ellas en el ámbito de este trabajo: el conjunto de datos de entrenamiento utilizado contiene 600.000 palabras *silver-standard* y 100.000 palabras *gold-standard*, en comparación con el corpus de 50.000 palabras utilizado anteriormente. El objetivo de este aumento del corpus era estudiar la utilidad de añadir datos *silver-standard* anotados automáticamente, junto con los datos *gold-standard* anotados manualmente. Este conjunto de datos se basa en TimeBank y en AQUAINT[1]. De forma adicional, se desarrolló un corpus *platinum* para la evaluación, anotado manualmente por expertos en el campo y basado en textos nuevos respecto a TempEval y TempEval-2.

En TempEval-3 se plantean 3 tareas principales: extracción y normalización de expresiones temporales (tarea A), extracción y clasificación de eventos (tarea B) y anotación de relaciones (tarea C). Como novedad respecto a las ediciones anteriores, se plantea la tarea de extremo a extremo, partiendo de los documentos del corpus y extrayendo expresiones temporales, eventos y relaciones en una única ejecución. No obstante, al igual que ocurría con las tareas de TempEval-2, para el ámbito de este trabajo sólo resultan de interés las tareas A y B. La tarea A sigue las mismas directrices que su homónima en TempEval-2, mientras que la tarea B consiste únicamente en la identificación de los eventos y su clasificación, pudiendo tomar el atributo `CLASS` los valores definidos en el estándar de anotación TimeML (Saurı et al., 2006): `REPORTING`, `PERCEPTION`, `ASPECTUAL`, `I-STATE`, `STATE` y `OCCURRENCE`.

2.2.2. Extracción de Eventos y Expresiones Temporales en el dominio clínico

En el marco de las conferencias i2b2, orientadas al avance de las líneas de investigación en el campo del Procesado del Lenguaje Natural dentro del dominio clínico, se planteó una tarea de extracción de conceptos en su edición de 2010 (Uzuner et al., 2011). Esta tarea consistía en la identificación y extracción de problemas médicos, tratamientos y pruebas diagnósticas den-

[1] https://doi.org/10.35111/pcbv-jq63

tro de informes clínicos de pacientes. En la edición de 2012 se amplía el alcance abordando la extracción de relaciones temporales en textos clínicos mediante una competición (Sun, Rumshisky, y Uzuner, 2013b), teniendo como objetivo la generación de líneas de tiempo sobre informes de alta de pacientes. Dado que se trató de la primera aproximación a la tarea que se realizaba sobre el dominio clínico, fue necesario construir un nuevo corpus con anotaciones temporales sobre informes de alta (Sun, Rumshisky, y Uzuner, 2013a).

El corpus se basa en el estándar de anotación TimeML, al igual que las conferencias SemEval expuestas anteriormente, aunque requiriendo pequeñas modificaciones. La extracción de eventos clínicos se basa en el esquema propuesto en la edición de 2010 de i2b2, contemplando como eventos todos los conceptos clínicos, y añadiéndose en esta edición también los departamentos clínicos (por ejemplo, "cardiología"), por resultar de interés a la hora de generar las líneas de tiempo de los pacientes. Las tareas de extracción de eventos y expresiones temporales en textos clínicos son muy similares a sus homónimas a nivel general, siendo necesario adaptar únicamente las dosis de medicamentos y las expresiones de frecuencia que se utilizan ampliamente en el vocabulario clínico, generalmente relativas a las frecuencias de las dosis. Dado que el corpus i2b2 es el utilizado en el desarrollo del presente trabajo, se describirá en profundidad en la sección 2.3.

En la conferencia SemEval 2015 se planteó nuevamente una tarea de TempEval, aunque alejándose de las anteriores ediciones al focalizarse en el dominio clínico. La tarea, denominada Clinical TempEval (Bethard et al., 2015), tiene como objetivo la extracción de líneas de tiempo sobre documentos de texto del dominio clínico, suponiendo una convergencia entre las tres primeras tareas de TempEval y la competición i2b2 del año 2012. Clinical TempEval se basa en el corpus THYME (Styler IV et al., 2014), compuesto por 600 informes clínicos de pacientes con cáncer de colon de la Clínica Mayo, anonimizados de forma manual para evitar datos que pudieran permitir la identificación de los pacientes en cuestión, sin afectar a las expresiones temporales ni a los eventos. Las anotaciones se hicieron basadas en una extensión del estándar ISO-TimeML (Pustejovsky et al., 2010), añadiendo nuevos tipos de expresiones temporales, como `PREPOSTEXP` para expresiones como "postoperatorio" (*postoperative*), y nuevos atributos para los eventos, como (`GRADE`) para cuantificar el grado de eventos como el dolor

o las nauseas.

Clinical TempEval consistió en 9 tareas, agrupadas en 3 categorías: identificación de expresiones temporales (`TIMEX3`), identificación de eventos (`EVENTs`) e identificación de relaciones temporales entre eventos y expresiones temporales. En el ámbito de este trabajo sólo resultan de interés las dos primeras categorías, al igual que en los casos anteriores, por lo que son las únicas que se describen a continuación:

- Identificación de `TIMEX3`: contempla la detección de los intervalos de texto (*spans*) comprendidos por las expresiones en la subtarea denominada *Time Spans* (TS); así como su clasificación entre las clases `DATE`, `TIME`, `DURATION`, `QUANTIFIER`, `PREPOSTEXP` y `SET`, en la subtarea *Time Attributes* (TA).

- Identificación de `EVENTs`: contempla la detección de los intervalos de texto (*spans*) comprendidos por los eventos en la subtarea denominada *Event Spans* (ES); así como los valores de sus atributos en la subtarea *Event Attributes* (EA):

  - Modalidad contextual (`Contextual Modality`): `ACTUAL`, `HEDGED`, `HYPOTHETICAL` o `GENERIC`
  - Grado (`DEGREE`): `MOST`, `LITTLE` o `N/A`
  - Polaridad (`POLARITY`): `POS` o `NEG`
  - Tipo (`TYPE`): `ASPECTUAL`, `EVIDENTIAL` o `N/A`

Las métricas utilizadas para evaluar los sistemas presentados a cada una de las categorías fueron la precisión (*precision*, P), la cobertura (*recall*, R) y el valor F1. Tomando S como el conjunto de predicciones del sistema y H como el conjunto de anotaciones del sistema, pueden definirse las métricas anteriores mediante las siguientes fórmulas (Bethard et al., 2015):

$$P = \frac{|S \cap H|}{|S|} \qquad R = \frac{|S \cap H|}{|H|} \qquad F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{2.1}$$

Se plantearon 2 sistemas basados en reglas como referencia para comparar los resultados de los sistemas participantes en la tarea: *memorize*, un sistema para todas las subtareas de extracción de eventos y expresiones temporales, y algunas de las tareas de extracción de relaciones; y *closest*, para

una de las subtareas de extracción de relaciones, no cubierta por el sistema *memorize*.

El sistema *memorize* se entrena utilizando todas las expresiones etiquetadas como `EVENT` o `TIMEX3` en el conjunto de datos de entrenamiento. Se buscan todas las coincidencias exactas de estas expresiones en el conjunto de datos, de forma que se conservan aquellas expresiones que se etiqueten como `EVENT` o `TIMEX3` en al menos un 50 % de sus instancias, asignándose entonces como etiqueta y atributos aquellos con mayor frecuencia entre dichas instancias. Los valores conservados durante el entrenamiento se utilizan para predecir sobre el conjunto de datos de test, buscando nuevamente las coincidencias exactas de las expresiones en los textos, y asignándoles la etiqueta y los valores de atributos memorizados por el sistema.

Tres equipos de investigación presentaron soluciones en la primera edición de Clinical TempEval:

BluLab El equipo compuesto por investigadores de la Universidad de Estocolmo y la Universidad de Utah participó en todas las tareas propuestas, utilizando clasificadores supervisados (Velupillai et al., 2015). En las subtareas correspondientes a la extracción de eventos (ES, EA) y expresiones temporales (TS, TA) recurren a Máquinas de Vectores Soporte (*Support Vector Machines*, SVM) para realizar una clasificación de cada *token*. Los clasificadores de intervalos de texto (*spans*) de las subtareas ES y TS generan representaciones acorde al formato IOB (Ramshaw y Marcus, 1995), de forma que el primer *token* de un fragmento de texto a etiquetar (en este caso, como evento o expresión temporal) recibe la etiqueta B-*class* (*Begin*), siendo *class* la clase a la que pertenece el fragmento; los siguientes *tokens* pertenecientes al mismo fragmento de texto reciben la etiqueta I-*class* (*Inside*), y todos aquellos *tokens* que no pertenecen a ningún evento o expresión temporal reciben la etiqueta O (*Outside*).

Para la subtarea ES se entrena un único clasificador IOB, haciendo uso de rasgos léxicos, incluyéndose también como rasgos de ventana, teniendo en cuenta los rasgos de los 2 *tokens* anteriores y posteriores a cada *token* a analizar, seguido de un clasificador por cada posible valor de la subtarea EA. En el caso de la subtarea TS, se entrena un clasificador IOB por cada posible valor del atributo `type` de la subtarea TA, haciendo uso de rasgos léxicos como los propios *tokens*, los

mismos prescindiendo de sus 2 últimos caracteres, etiquetas gramaticales (*part-of-speech*, POS), rasgos booleanos (si el *token* es de tipo numérico, comienza por mayúscula o está completamente en minúscula) y rasgos de ventana, sin especificarse el tamaño de la misma; así como información de un *gazetteer* basado parcialmente en una versión adaptada de HeidelTime (Strötgen y Gertz, 2013), un sistema basado en reglas. Para la subtarea TA se sigue un planteamiento similar que para TS, variando el tamaño de ventana de los *tokens* y la información del *gazetteer* en función para cada tipo de TA, sin especificarse en el artículo los valores anteriores.

KPSCMI El equipo de Kaiser Permanente South Carolina participó únicamente en las subtareas de expresiones temporales (TS y TA), recurriendo a una versión extendida de HeidelTime con clasificadores supervisados, sin especificarse más detalles sobre la solución en el artículo de Clinical TempEval (Bethard et al., 2015).

UFPRSheffield El equipo compuesto por investigadores de la Universidad Federal de Paraná y la Universidad de Sheffield participó únicamente en las subtareas TS y TA, comparando dos aproximaciones: sistemas basados en reglas y basados en SVM (Tissot et al., 2015).

Para la aproximación basada en reglas se desarrolla un sistema, denominado HYNX, haciendo uso de GATE (Cunningham et al., 2011). El sistema ejecuta un conjunto de reglas y *scripts*, ejecutadas de forma jerárquica como parte de un *pipeline* de extracción de información, dividido en tres módulos: preprocesado de texto, identificación de `TIMEX3` y normalización de `TIMEX3`. Estos módulos se encargan de identificar y normalizar conceptos temporales, encontrando y agrupando los *tokens* que puedan forman expresiones temporales más complejas para, posteriormente, normalizar sus rasgos.

La aproximación basada en SVM también se apoya en GATE, que integra una modificación de LibSVM (Chang y Lin, 2011) para facilitar el prototipado en tareas de identificación y clasificación de entidades nombradas. Se recurre a 2 SVM para cada posible clase, una para la identificación del inicio de la entidad (en este caso, la expresión temporal) y otra para la identificación del final de la misma. A continuación, se procesa la información obtenida para eliminar las etiquetas de inicio

y fin desparejadas, ya que no permiten formar entidades, y se filtran las expresiones cuya longitud en cuanto a número de palabras no estuviera presente en el conjunto de entrenamiento. Por último, en el caso de solapamientos entre varias clases para una misma entidad se selecciona el resultado final entre todas las clases mediante el grado de confianza, eligiendo el candidato de mayor grado. Tras ello, se realiza un segundo test de confianza para eliminar entidades con un grado de confianza débil.

De los sistemas anteriores, se presentaron los resultados de 3 ejecuciones de BluLab y KPSCMI, y 7 ejecuciones de UFPRSheffield: 2 ejecuciones de la aproximación basada en SVM y 5 de la basada en reglas (HYNX). En la tarea de detección de *spans* de `TIMEX3` (TS), el sistema BlueLab obtuvo los mejores resultados de precisión y valor F1, mientras que el mejor valor de cobertura (*recall*) lo obtuvo el sistema HYNX. Estos sistemas obtuvieron los mejores resultados en las mismas métricas para la tarea TA, que medía la detección de los *spans* junto con la clasificación de las expresiones temporales identificadas. En cuanto a las tareas de identificación y clasificación de `EVENTs` (ES y EA), BlueLab obtuvo los mejores resultados, siendo el único sistema presentado a las tareas, superando al sistema de referencia *memorize*.

En la conferencia SemEval 2016 volvió a proponerse Clinical TempEval como una de sus tareas (Bethard et al., 2016), con las mismas características que la edición anterior, incluyendo los mismos sistemas de referencia y procedimientos de evaluación, a excepción de la evaluación de los atributos de los `EVENTs`, que pasan de contemplarse como una simple clasificación general a evaluarse cada atributo de forma independiente. En esta edición aumentó notablemente el número de participantes, pasando de 3 equipos de investigación y 13 ejecuciones en la edición de 2015, a 14 equipos y 40 ejecuciones en 2016. Los sistemas presentados fueron los siguientes:

Brundlefly Propone dos aproximaciones, una de aprendizaje no supervisado basada en redes de neuronas recurrentes, *word embeddings* y regresión logística; y una de aprendizaje supervisado basada en DeepDive[2] (Fries, 2016).

[2] http://deepdive.stanford.edu

CDE-IIITH Propone dos aproximaciones, una basada en redes de neuronas profundas, y otra en Conditional Random Fields (CRF) y SVM (Chikka, 2016).

Cental (Hansart et al., 2016) proponen un sistema basado en CRF y recursos léxicos como *Unified Medical Language System* (UMLS) (Bodenreider, 2004), que recoge terminología del dominio biomédico, y reglas lingüísticas implementadas mediante el *framework* Unitex (Paumier, 2003).

GUIR Propone un sistema basado en CRF y regresión logística con rasgos léxicos, morfológicos, sintácticos, de dependencias y específicos del dominio, combinados con reglas para el reconocimiento de patrones (Cohan, Meurer, y Goharian, 2016).

HITACHI Propone un sistema ensamblado compuesto por un sistema basado en reglas y modelos de aprendizaje automático, recurriendo a rasgos léxicos, sintácticos y morfológicos (Sarath, Manikandan, y Niwa, 2016).

KULeuven-LIIR (Leeuwenberg y Moens, 2016) proponen un modelo de aprendizaje automático basado en cTAKES-Temporal (Lin et al., 2016).

LIMSI (Grouin y Moriceau, 2016) proponen un sistema basado en CRF con rasgos léxicos, morfológicos y de *clusters* de palabras, junto con el sistema basado en reglas HeidelTime (Strötgen y Gertz, 2013).

LIMSI-COT Propone dos aproximaciones, la primera basada en SVM con rasgos léxicos, sintácticos, estructurales y de UMLS; y la segunda basada en la misma arquitectura sustituyendo los rasgos léxicos por *word embeddings* (Tourille et al., 2016).

ULISBOA (Barros et al., 2016) proponen un sistema basado en SVM con rasgos léxicos y morfológicos, junto a un sistema basado en reglas que extiende las incluidas en Stanford CoreNLP (Manning et al., 2014).

UtahBMI Propone dos aproximaciones, la primera basada en CRF y la segunda en SVM, utilizando ambas rasgos léxicos, morfológicos, sintácticos, formas de palabras, patrones de caracteres, n-gramas y *gazetteer* (Khalifa, Velupillai, y Meystre, 2016).

UTA-MLNLP Propone una aproximación basada en una red de neuronas con rasgos de ventana, comparando en dos ejecuciones diferentes una ventana de tamaño $w = 4$ y otra $w = 5$ (Li y Huang, 2016).

UTHealth Propone una aproximación basada en SVM lineales y Modelos de Markov Ocultos (*Hidden Markov Models*, HMM), recurriendo a rasgos léxicos, morfológicos, sintácticos, de discurso y representaciones de palabras (Lee et al., 2016).

VUACLTL (Caselli y Morante, 2016) proponen una aproximación basada en CRF con rasgos morfosintácticos, léxicos, UMLS y de DBpedia (Bizer et al., 2009).

Los mejores resultados de las tareas de reconocimiento y clasificación de expresiones temporales los consigue el sistema de UTHealth, obteniendo en ambas los valores más altos de cobertura (*recall*) y valor F1. El mejor valor de precisión lo obtiene el sistema presentado por LIMSI, aunque su valor F1 (el utilizado para ordenar la clasificación) está 0,18 puntos por debajo del mejor.

En cuanto a las tareas de identificación y clasificación de eventos, el sistema UTHealth vuelve a obtener los mejores resultados, en este caso en todas las métricas, en la detección de *spans* y en la clasificación de todos los atributos de los eventos (`MODALITY`, `DEGREE`, `POLARITY` y `TYPE`).

En comparación con la anterior edición de SemEval, los resultados del mejor sistema en las tareas de reconocimiento y clasificación de `TIMEX3` superan al anterior estado del arte, con un valor F1 en la detección de *spans* de 0,795 conseguido por el sistema de UTHealth frente a 0,725 del sistema de BluLab en la edición anterior; y un valor F1 en la clasificación de expresiones temporales de 0,772 frente a 0,709 para los mismos sistemas. Respecto a las tareas de detección y clasificación de `EVENTs`, en la detección de *spans* el sistema UTHealth obtiene un valor F1 de 0,903 frente a 0,875 conseguido por BluLab; en la clasificación del atributo `MODALITY` consigue un valor F1 de 0,855 frente a 0,824; en la clasificación del atributo `DEGREE` un 0,899 frente a 0,87; en la clasificación del atributo `POLARITY` un 0,887 frente a 0,857; y en la clasificación del atributo `TYPE` un 0,882 frente a 0,823. Por tanto, se puede concluir que en la tarea Clinical TempEval de SemEval 2016 se consiguen superar los mejores resultados en todas las categorías respecto a SemEval 2015.

La última edición de Clinical TempEval se desarrolló como parte de la conferencia SemEval 2017 (Bethard et al., 2017). Esta edición conserva las mismas tareas que las dos anteriores pero, en lugar de tener como objetivo únicamente la extracción de líneas temporales, pretende evaluar la capacidad de generalización de sistemas entrenados para extraer líneas temporales sobre un dominio concreto, como es el caso de los informes de alta de pacientes de cáncer de colon utilizados en las dos ediciones anteriores, evaluando sobre un conjunto de datos de dominio diferente, en este caso sobre informes de alta de pacientes de tumor cerebral.

Participaron 11 equipos de investigación, con un total de 28 ejecuciones de sus sistemas. Los equipos presentados fueron los siguientes:

GUIR Propone una combinación de CRF, reglas y árboles de decisión ensamblados, utilizando como rasgos n-gramas, palabras, formas de palabras, *clusters* de palabras, etiquetas gramaticales (*part-of-speech*, POS), árboles sintácticos y de dependencias, roles semánticos y tipos de conceptos extraídos de UMLS (MacAvaney, Cohan, y Goharian, 2017).

Hitachi Propone una combinación de CRF, reglas, redes de neuronas y árboles de decisión ensamblados, utilizando rasgos como n-gramas, formas de palabras, *word embeddings*, tiempos verbales, encabezados de secciones del documento, y *sentence embeddings* (Sarath, Manikandan, y Niwa, 2017).

KULeuven-LIIR Propone una combinación de SVM y perceptrones estructurados, haciendo uso de rasgos como las palabras y las etiquetas POS (Leeuwenberg y Moens, 2017). Además, para la adaptación del sistema al nuevo dominio, asignaron pesos más altos a los datos de entrenamiento de pacientes de tumores cerebrales, y añadieron la representación de términos desconocidos al vocabulario de entrada.

LIMSI-COT Propone una combinación de redes de neuronas recurrentes, haciendo uso de *word embeddings* y de *character embeddings*, y SVM con rasgos como palabras y etiquetas POS (Tourille et al., 2017). Para la adaptación de dominio de los datos, proponen impedir la modificación de los *word embeddings* preentrenados, además de añadir la representación de términos desconocidos al vocabulario de entrada.

NTU-1 Propone una combinación de SVM y CRF, utilizando rasgos como n-gramas, etiquetas POS, formas de palabras, entidades nombradas, árboles de dependencia y tipos de conceptos de UMLS (Huang et al., 2017).

ULISBOA Propone una combinación de CRF y reglas, recurriendo a rasgos como n-gramas, palabras, etiquetas POS y tipos de conceptos de UMLS (Lamurias et al., 2017).

XJNLP Propone una combinación de reglas, SVM y redes de neuronas recurrentes y convolucionales, usando rasgos como palabras, *word embeddings* y tiempos verbales (Long et al., 2017).

Todos los sistemas anteriores tienen en común la utilización de algún clasificador basado en aprendizaje supervisado, con rasgos como n-gramas, etiquetas POS, y tipos de conceptos extraídos de UMLS. También se presentaron cuatro equipos que no proporcionaron una descripción de sus sistemas: WuHanNLP, UNICA, UTD e IIIT.

En las tareas de identificación de `TIMEX3`, el mejor resultado lo obtiene el sistema de GUIR con un valor F1 que oscila entre 0.51 y 0.59, aproximadamente 2 décimas inferior al mejor resultado registrado en Clinical TempEval 2016. Los resultados de las tareas de extracción de `EVENTs` son similares, obteniendo aproximadamente un valor F1 de 0,70 en todas las subtareas, nuevamente con una diferencia de 2 décimas respecto a los resultados conseguidos en la edición de 2016. La principal hipótesis planteada en (Bethard et al., 2017) para la notable reducción de las métricas respecto a la edición anterior es la evaluación de los sistemas sobre un corpus de características diferentes al conjunto de datos de entrenamiento. El hecho de que estos conjuntos de datos pertenezcan a dominios diferentes altera la composición de los mismos, tanto en términos de vocabulario no comunes entre ambos, como en la frecuencia de las entidades, como es el caso de la expresión temporal "*overnight*", que aparece 148 veces en el corpus de pacientes de tumor cerebral, frente a 11 en el de cáncer de colon, pudiendo alterar los resultados.

2.3. Colección de datos

El estado del arte actual se basa principalmente en el corpus THYME (Styler IV et al., 2014) por ser el más reciente, y por permitir evaluar los

sistemas sobre informes de pacientes de tumor cerebral, un subdominio diferente al utilizado para el entrenamiento, donde se emplean informes de pacientes de cáncer de colon. El otro corpus de relevancia en la tarea es i2b2 (Sun, Rumshisky, y Uzuner, 2013a), recibiendo menos atención por ser ligeramente más antiguo, y por no hacer distinciones entre las patologías de los pacientes para estructurar los conjuntos de entrenamiento y evaluación. Dado que no se dispone de acceso al corpus THYME, los experimentos del presente trabajo se basan en el corpus i2b2, que contiene 310 informes de alta de pacientes provenientes de las organizaciones Partners Healthcare y Beth Israel Deaconess Medical Center, compuestos por un total de 178.000 *tokens*.

El corpus contempla dos grupos de anotaciones, uno correspondiente a los eventos y expresiones temporales, y otro para las relaciones temporales entre las anteriores. Los sistemas presentados en este documento se evalúan únicamente sobre los eventos y expresiones temporales, por lo que no se profundizará en definir las relaciones.

La etiqueta `EVENT` sirve para identificar aquellas anotaciones correspondientes a eventos clínicos relevantes, como pueden ser:

- Conceptos clínicos: problemas (`PROBLEM`), como "*HIV positive*"; pruebas (`TEST`), como "*a CT scan*"; y tratamientos (`TREATMENT`), como "*H2 blockers*"; tal y como se definen en (Uzuner et al., 2011).
- Departamentos clínicos (`CLINICAL_DEPT`): como la Unidad de Cuidados Intensivos ("*the Medical Intensive Care Unit*") o el quirófano ("*the operating room*").
- Evidencias (`EVIDENTIAL`): todos aquellos eventos que determinen la fuente de la información descrita (por ejemplo, "*presented*" y "*show*").
- Ocurrencias (`OCCURRENCE`): todos aquellos eventos que le sucedan al paciente, como el ingreso ("*admission*") o el alta ("*discharge*").

Cada `EVENT` tiene tres atributos: tipo (`type`), que determina a qué categoría de las anteriores pertenece; polaridad (`polarity`), que determina si el evento es positivo (`POS`) o negativo (`NEG`); y modalidad (`modality`), que indica si el evento ha sucedido (`FACTUAL`), se ha propuesto (`PROPOSED`), se ha mencionado como una condición (`CONDITIONAL`) o si es posible o hipotético (`POSSIBLE`).

Por otra parte, la etiqueta TIMEX3 sirve para identificar las expresiones temporales, considerándose como tal las fechas (DATE), horas (TIME), duraciones (DURATION) y frecuencias (FREQUENCY) presentes en el texto. Cada TIMEX3 tiene tres atributos: tipo (type), que determina la categoría a la que pertenece de entre las expuestas anteriormente; valor (value), que determina su valor normalizado siguiendo el estándar ISO-8601; y modificador (modifier), que determina si un valor temporal es exacto (NA) o, por el contrario, se trata de algún tipo de aproximación (APPROX), un valor superior (MORE) o inferior (LESS) al contenido en la expresión, o está al inicio (START), fin (END) o en medio (MIDDLE) de un periodo de tiempo determinado, como puede ser "a principios de año" o "a mediados de septiembre".

En promedio, cada informe de alta contiene 86.6 EVENTs y 12.4 TIMEX3.

Capítulo 3

Sistemas propuestos

En este capítulo se expone en profundidad la fase de experimentación del proyecto, describiendo las diferentes arquitecturas propuestas para abordar el problema, así como la evolución de las mismas hasta llegar a los sistemas a evaluar en el capítulo 4.

3.1. Sistemas de referencia

Tomando como punto de partida la tarea 12 de la conferencia SemEval 2017 (Bethard et al., 2017), se busca entre los sistemas presentados a dicha tarea una arquitectura que sirva como base para el desarrollo del sistema a proponer. La finalidad de dicha tarea era la extracción de relaciones temporales entre eventos en documentos clínicos de distintos dominios, entrenando con documentos de pacientes de cáncer de colon y evaluando con documentos de pacientes de tumor cerebral.

Entre todos los sistemas presentados a SemEval 2017, los dos sistemas con mejores resultados de la competición fueron los siguientes:

- (MacAvaney, Cohan, y Goharian, 2017): Combinación de CRF (*Conditional Random Fields*) (Lafferty, McCallum, y Pereira, 2001), reglas y conjuntos de árboles de decisión. Entre los diferentes rasgos que utiliza se incluyen n-gramas, tokens, formas de palabras, clusters de palabras, *word embeddings*, etiquetas gramaticales (*Part-Of-Speech*, POS), árboles sintácticos y de dependencia, roles semánticos y tipos de conceptos de UMLS. Este planteamiento obtuvo los mejores resultados de valor F1 para el reconocimiento de expresiones temporales, a excepción de la

clasificación de las mismas tras haber detectado su intervalo de texto (*span*), donde obtuvo el segundo mejor resultado.

- (Leeuwenberg y Moens, 2017): Combinación de SVM (*Support Vector Machines*) (Boser, Guyon, y Vapnik, 1992) y perceptrones estructurados, utilizando como rasgos los *tokens* y sus etiquetas POS. Además, para la fase de adaptación se aumenta el peso de los datos de entrenamiento relativos a tumores cerebrales, y se representan palabras desconocidas en el vocabulario de entrada. Este planteamiento tuvo el mejor resultado en la detección del intervalo de las expresiones temporales junto con su clase en la fase de aprendizaje no supervisado, y el segundo puesto en la detección del intervalo únicamente.

El segundo sistema se considera de menor complejidad tanto a nivel de arquitectura como de preprocesamiento de los datos, por lo que se utiliza como base para los sistemas a plantear. A pesar de que los sistemas presentados a SemEval 2017 se desarrollaron y evaluaron mediante el corpus THYME (Styler IV et al., 2014), a fecha de la redacción de este documento no se ha conseguido acceso al mismo, por lo que se ha recurrido al uso del corpus i2b2 (Sun, Rumshisky, y Uzuner, 2013a), el anterior estándar utilizado en esta línea de investigación. A pesar de utilizarse el corpus i2b2, se han conservado como referencias los sistemas de SemEval 2017 ante la posibilidad de conseguir acceso al corpus THYME eventualmente.

3.2. Procesado del corpus

Los datos del corpus requieren un preprocesado previo a su uso en los sistemas a desarrollar. Este preprocesado dependerá de la arquitectura del sistema en cuestión, pero siempre incluirá los *tokens* que componen el texto. En cuanto a los datos a predecir, se plantea la detección de los intervalos de texto (*spans*) de las entidades mediante etiquetadores IOB, distinguiéndose las etiquetas `I-eve` y `B-eve` para eventos, `I-tim` y `B-tim` para expresiones temporales, y la etiqueta `O` para todos aquellos *tokens* que no pertenezcan a ninguna entidad. Las etiquetas `B` representan el primer *token* de la entidad del tipo correspondiente (`eve` o `tim`), mientras que las etiquetas `I` representan aquellos *tokens* que forman parte de dicha entidad.

Además de los intervalos, cada tipo de entidad tiene unos atributos característicos que también deben detectarse. En el caso de los eventos, se

Sistema	TIMEX3 Spans	TIMEX3 Atributos	EVENT Spans	EVENT Atributos	Método
KU	0,61	0,60	0,62	0,60	SVM
KU_w1	0,80	0,75	0,81	0,72	SVM, w=1
KU_w2	0,82	0,73	0,81	0,72	SVM, w=2
KU_w3	0,81	0,72	0,80	0,75	SVM, w=3
KU_w1*	0,77	0,36	0,81	0,72	SVM, w=1
KU_w1**	0,77	0,71	0,81	0,72	SVM, w=1

Tabla 3.1: Resultados de exactitud (*accuracy*) del primer sistema durante el entrenamiento.

debe predecir el valor de los atributos `type`, que determina el tipo de evento, `polarity`, que define si el evento es positivo o negativo, y `modality`; mientras que para las expresiones temporales deben predecirse los atributos `type`, que determina el tipo de expresión temporal, `value`, que recoge el valor de la expresión en formato estándar, y `modifier`.

Algunos de los modelos planteados no permiten predicción multiclase, como es el caso de las SVM. La predicción de atributos es de tipo multiclase, por lo que para los modelos no compatibles se recurrirá a unificar los atributos correspondientes en una única clase. Al tratarse de atributos de tipo *string*, se pueden concatenar introduciendo un caracter que permita su posterior separación para la escritura en los ficheros de anotación, como puede ser el caracter "`|`". Por ejemplo, en el caso de los eventos, la clase a predecir sería `type|polarity|modality`.

3.3. Primera aproximación

Tal y como se ha expuesto en la sección 3.1, el primer sistema a plantear estará basado en la aproximación de (Leeuwenberg y Moens, 2017), debido a la facilidad para replicar su arquitectura y el preprocesamiento necesario para la uso de los datos.

Los rasgos a utilizar en este sistema serán las palabras, etiquetas POS, y estos mismos campos de los tokens anteriores/siguientes para un tamaño de ventana (w) determinado. A pesar de que en el artículo original se recomienda un tamaño de ventana de entre 3 y 5 tokens para el corpus THYME, en la práctica se ha observado que para el corpus i2b2 no se producen mejoras para un tamaño de ventana superior a 1, aunque el tiempo de ejecución au-

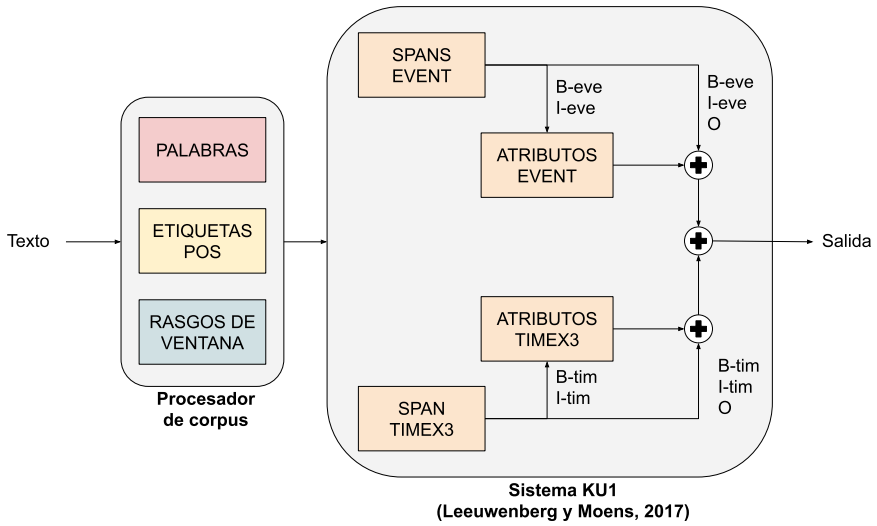

Figura 3.1: Arquitectura del sistema KU1 (Leeuwenberg y Moens, 2017)

menta de forma exponencial. Por tanto, se utilizará un tamaño de ventana de 1. Estos resultados se pueden observar en la tabla 3.1, en los sistemas con nombre `KU_wn`, siendo `n` el tamaño de la ventana.

El sistema, que se denominará `KU1` se compone de 4 SVM, una para cada una de las siguientes funciones: predicción de *spans* de eventos (ES) y expresiones temporales (TS), y predicción de atributos de eventos (EA) y expresiones temporales (TA). La arquitectura queda plasmada en la figura 3.1.

En la etapa de preprocesamiento todos los *tokens* se convierten a minúsculas, y todos los dígitos se sustituyen por una representación unificada para facilitar su detección, tal y como se propone en (Leeuwenberg y Moens, 2017; Miller et al., 2015). En este caso, se utilizará el *token* "5"para ese fin por lo que, por ejemplo, el *token 1995* se sustituye por *5555*, y el *token 1995-04-23* se sustituye por *5555-55-55*.

3.4. Segunda aproximación

Como se expondrá en el capítulo 4, durante la evaluación del sistema se observa una exactitud de 0 para la predicción del campo `val` de las expresiones temporales debido a la sustitución de los dígitos por la representación

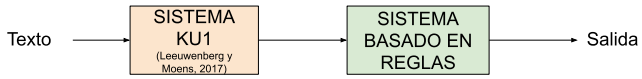

Figura 3.2: Arquitectura del sistema KU2

unificada, por lo que resulta necesario plantear una modificación en la arquitectura para tratar de solventar esta incidencia. Este error se produce en su mayoría en expresiones del tipo `DATE`, al ser las más frecuentes y las únicas, junto con las expresiones de tipo `TIME` que en el texto aparecen como dígitos.

Debido a lo anterior, se plantea un nuevo sistema denominado `KU2`, cuya arquitectura será idéntica a la del sistema `KU1`, pero añadiendo en la salida un sistema basado en reglas, tal y como se representa en la figura 3.2.

El sistema basado en reglas añadido servirá para predecir los valores de los *tokens* cuyo tipo se haya determinado como `DATE` o `TIME` por parte del sistema `KU1`. También se añade una modificación sobre el preprocesamiento de los datos, aplicando únicamente la representación unificada de los dígitos sobre el valor del atributo `value` de las expresiones de los tipos anteriores, de forma que permanezcan intactos los *tokens* extraídos del texto. Esto se debe a que, a la hora de predecir, el sistema no dispondrá de información para hacer el preprocesado de los *tokens* sólo en estos casos, y aplicarlo sobre todos los *tokens* que contengan dígitos puede llevar a una pérdida de información, al imposibilitar recuperar el valor original de los mismos después de su conversión a la representación unificada.

Las reglas se han evaluado de forma que se detecten en primer lugar los formatos de fecha más restrictivos mediante expresiones regulares. El formato más restrictivo presente en el corpus es el ISO-8601, que coincide con el formato objetivo del campo `value` de las expresiones de tipo `DATE` y `TIME`. También se han identificado otros formatos para las fechas en el corpus: `MM/DD/YYYY` y `DD/MM/YYYY`, así como sus derivados omitiendo cifras (por ejemplo, *23/4/95*). Estos formatos se detectan en el orden anterior, ya que ambos son igual de restrictivos y pueden dar lugar a casos ambiguos, como *09/06/1999*. Dado que el primero de los formatos anteriores es mucho más frecuente en el corpus, será el primero en intentar detectarse, y el que se asignará en caso de ambigüedad.

Las fechas, como se ha expuesto anteriormente, pueden mostrarse de

forma abreviada, aunque en el atributo `value` deben mostrarse completas. Para días y meses será suficiente con añadir un *0* delante del valor del texto, en caso de que este tenga un solo dígito. En cuanto a los años, cuya forma abreviada se compone de los dos últimos dígitos (por ejemplo, *99* en lugar de *1999*), se planteó en primer lugar la posibilidad de tomar los dos primeros dígitos de la fecha de ingreso, pero esta en algunos documentos también se presentaba en formato abreviado. Por ello, partiendo de que los documentos del corpus más antiguos datan de la década de *1980*, y los más nuevos de la década de *2010*, se añadirán los dos primeros dígitos de forma que si los dos últimos tienen un valor inferior a *20* (*2020*), los primeros dígitos tomarán el valor *20* por tratarse de una fecha del siglo XXI. Si, por el contrario, el valor de los dos dígitos tiene un valor superior a *20*, se considerará una fecha del siglo XX, por lo que los dos primeros dígitos tomarán el valor *19*.

Observando varios ficheros del corpus elegidos al azar, se planteó la hipótesis de que todas las fechas de un documento siguen el mismo formato, por lo que en la primera versión de esta aproximación se detecta el formato de la primera fecha del documento (la fecha de ingreso del paciente) para saber qué formato comprobar para el resto y evitar evaluar todos los formatos para todas las expresiones, evitando también posibles ambigüedades.

No obstante, un análisis más exhaustivo del corpus permitió encontrar casos donde la fecha de ingreso se encontraba en formato ISO-8601, mientras que las fechas del cuerpo del documento se encontraban en formato diferente, por lo que para una segunda versión del sistema se comprobaban todos los posibles formatos para cada fecha encontrada.

Esta segunda versión del sistema también contempla el aumento del tamaño de ventana del módulo `KU1`, determinando tras diversas pruebas que con una ventana $w = 2$ se obtienen mejores resultados que con la ventana original ($w = 1$), y que con tamaños de ventana superiores no se consiguen mejoras significativas. Estos resultados se expondrán en el capítulo 4.

3.5. Detección de *spans* con CRF

La siguiente modificación a introducir en la arquitectura se apoya en el caracter secuencial del texto para detectar los intervalos de los eventos y las expresiones temporales. Al observarse que la mayoría de los sistemas presentados a la competición del corpus i2b2 (Sun, Rumshisky, y Uzuner,

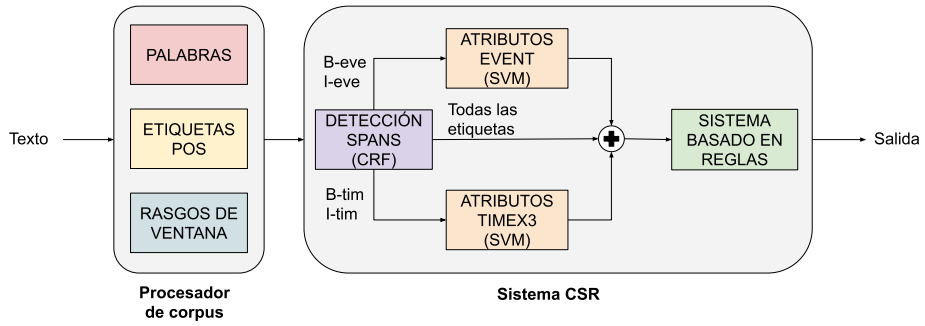

Figura 3.3: Arquitectura del sistema CSR

2013b) se componen de CRF, SVM, o bien combinaciones de ambos, se decide plantear un nuevo sistema con una arquitectura similar al `KU2`, pero sustituyendo las SVM encargadas de la detección de los *spans* de los eventos (ES) y expresiones temporales (TS) por un único CRF, debido a que la etiquetación IOB se puede plantear como un problema de clasificación de secuencias. La arquitectura de este nuevo sistema, que se denomina `CSR`, se muestra en la figura 3.3.

Tras diversas pruebas se observa que, al igual que sucede con el sistema `KU2`, no se obtienen mejoras significativas para tamaños de ventana superiores a 2, por lo que el sistema `CSR` también utilizará una ventana $w = 2$ para sus rasgos de entrada. Inicialmente el sistema se ejecuta sin ningún tipo de regularización sobre el CRF.

Se plantea una segunda versión del sistema en la que sí se aplica regularización sobre el CRF, denominada *CSR**. Se aplica regularización para minimizar la función de pérdida y prevenir el sobreentrenamiento del modelo, utilizando regularización L1 (Lasso) para minimizar en torno a la mediana de la distribución de los datos, y regularización L2 (Ridge) para minimizar respecto a la media de los datos. Para encontrar los valores óptimos de los hiperparámetros encargados de la regularización L1 y L2, se recurre a una validación cruzada mediante Random Search y K-Fold.

Al igual que en los sistemas anteriores, los resultados de ambas versiones de la arquitectura *CSR* se expondrán en el capítulo 4.

3.6. Ingeniería de rasgos

Habiendo planteado dos arquitecturas diferentes para solucionar el problema, se procede a valorar diferentes rasgos que pueden añadirse para mejorar los resultados obtenidos en el mejor sistema de cada arquitectura. En primer lugar, se valorarán rasgos básicos dependientes del texto, para evaluar más tarde rasgos específicos del dominio de la tarea.

3.6.1. Rasgos dependientes del texto

Los primeros rasgos básicos dependientes del texto a valorar son los booleanos `word.istitle()` y `word.isupper()`, que comprueban si la palabra comienza o está escrita completamente en mayúsculas, respectivamente. El primero de estos rasgos produce unas ligeras mejoras (expuestas en el capítulo 4) en el modelo `KU2_w2` en los *spans* de los eventos y expresiones temporales, y en el campo `value` de las últimas. Si se incluye como rasgo de ventana, produce una mejora similar en los campos `type` y `modifier` de las expresiones temporales en este mismo sistema. Para el sistema `CSR*` no se produce una variación significativa, pero se conserva al mejorar el anterior sistema y no empeorar este. Por otra parte, el rasgo `word.isupper()` no produce mejoras significativas en ninguno de los sistemas, por lo que este rasgo se descarta para las siguientes evaluaciones.

También se evalúan como rasgos los prefijos y sufijos de los *tokens*, siendo necesario determinar los caracteres a abarcar en cada caso. Entre los sistemas participantes a la competición del corpus i2b2 (Sun, Rumshisky, y Uzuner, 2013b), se toma como referencia el que obtuvo los mejores resultados (Xu et al., 2013), que, a su vez, tiene una arquitectura similar al sistema `CSR*`. Entre todos los rasgos utilizados en este sistema (Xu et al., 2012), se encuentran rasgos derivados de prefijos y sufijos, incluyendo entre 2 y 7 caracteres al inicio y fin de cada *token*. Estos rasgos se evaluarán añadiendo los prefijos por longitudes de forma incremental, comenzando por los prefijos y sufijos de longitud 2, y añadiendo los de las siguientes longitudes a valorar. En el capítulo 4 pueden observarse las métricas obtenidas en cada caso, donde se concluye que para el sistema `KU2_w2` sólo deben incluirse prefijos y sufijos de longitudes 2 y 3, mientras que en el sistema `CSR*` se incluyen hasta la longitud 5. También se podrá concluir que estos rasgos pueden producir sobreentrenamiento si se utilizan como rasgos de ventana,

por lo que los únicos rasgos de ventana a utilizar serán los *tokens* y sus etiquetas gramaticales.

Finalmente, se plantea la inclusión de la representación unificada de los dígitos como rasgo independiente. De esta forma, además de contar con los *tokens*, el sistema contará a la entrada con la representación unificada de los dígitos cuando corresponda, mediante el rasgo `conflated token`, en lugar de sustituir el valor en los *tokens* originales como se planteaba en el sistema `KU1`, o de aplicarlo únicamente en el atributo `value` de las expresiones temporales como se había determinado mediante la evaluación del sistema. Este nuevo rasgo permite mejorar los sistemas `KU2_w2` y `CSR*` con los rasgos anteriores, comprobando que siguen obteniéndose los mejores resultados con las mismas longitudes de prefijos y sufijos.

3.6.2. Rasgos específicos del dominio

Tras haber evaluado una serie de rasgos simples dependientes del texto, se procede a incluir rasgos específicos del dominio de la tarea. Se plantea que la utilización del *Unified Medical Language System* (UMLS) (Bodenreider, 2004), en su versión 2020AA, podría permitir mejorar la clasificación del campo `type` para los eventos, por lo que se define el rasgo `UMLSsemtype` para recoger el tipo semántico de UMLS detectado para cada token. De todos los tipos semánticos disponibles, inicialmente se utilizarán, tal y como se propone en (MacAvaney, Cohan, y Goharian, 2017), `diagnostic procedure`, `disease or syndrome` y `therapeutic procedure`, que se corresponden respectivamente con los valores `test`, `problem` y `treatment` del atributo `type`.

A pesar de que el tipo semántico detectado por UMLS quede registrado por cada *token*, el análisis se realizará por oraciones mediante QuickUMLS (Soldaini y Goharian, 2016). QuickUMLS utiliza por defecto un umbral de similitud de 0,7 para reconocer expresiones, pero se ha observado que esto da algunos falsos positivos (ej: el verbo `increased` se reconoce como `increased ph`), por lo que se aumenta este umbral hasta 0,75.

3.7. Detección de *spans* sobre el atributo `type`

Tras haber determinado nuevos rasgos que permiten mejorar los resultados de los sistemas propuestos, se observa que, tanto para eventos como para expresiones temporales, el atributo `type` es el que presenta unas métricas

más distantes de los sistemas presentados la competición, tal y como se expone en el capítulo 4.

El atributo `type` es común a ambos tipos de entidades, pero no presenta solapamientos entre valores, por lo que se planteó la posibilidad de realizar el reconocimiento de *spans* directamente sobre los posibles valores del campo *type*, modificando las etiquetas IOB utilizadas de forma que no se diferencie entre entidades de tipo evento o expresión temporal, sino directamente por los subtipos de estas entidades (`DATE`, `TIME`, `FREQUENCY`, etc), tal y como se propone en (Gupta, Joshi, y Bhattacharyya, 2015). Una vez detectados estos subtipos, podrá inferirse si se trata de una entidad de tipo evento o de una expresión temporal.

Tras realizar una prueba con el sistema `CSR*`, por ser el que mejores métricas presenta entre los sistemas propuestos hasta este punto, se obtienen puntuaciones similares a las anteriores en la detección de *spans*, pero otros campos empeoran considerablemente sus métricas, por lo que queda descartado este planteamiento y no se incluirá en el proceso de evaluación.

3.8. Detección de *spans* basada en modelos de Deep Learning

Tras el buen resultado obtenido al plantear la detección de *spans* como un problema de clasificación de secuencias, se planteó extender el planteamiento sustituyendo el CRF por un modelo de Deep Learning, al tratarse de un enfoque que no fue abordado por ninguno de los sistemas de la competición del corpus i2b2 (Sun, Rumshisky, y Uzuner, 2013b). Estos modelos son más complejos, pudiendo permitir obtener mejores resultados, siendo estos menos dependientes del resultado de la ingeniería de rasgos y del preprocesamiento de los datos.

En esta sección se exponen las dos arquitecturas planteadas para este tipo de modelos: una basada en redes BiLSTM y otra basada en BERT.

3.8.1. Detección de *spans* mediante BiLSTM

El primer modelo de Deep Learning a probar como sustitución del CRF será una red de neuronas *Long Shot-Term Memory* (LSTM) bidireccional. Las redes LSTM (Hochreiter y Schmidhuber, 1997) son un tipo de redes neuronales recurrentes cuya unidad atómica no es la neurona, sino la celda

LSTM, una unidad de memoria con capacidad para mantener el contexto y modelar dependencias a largo plazo.

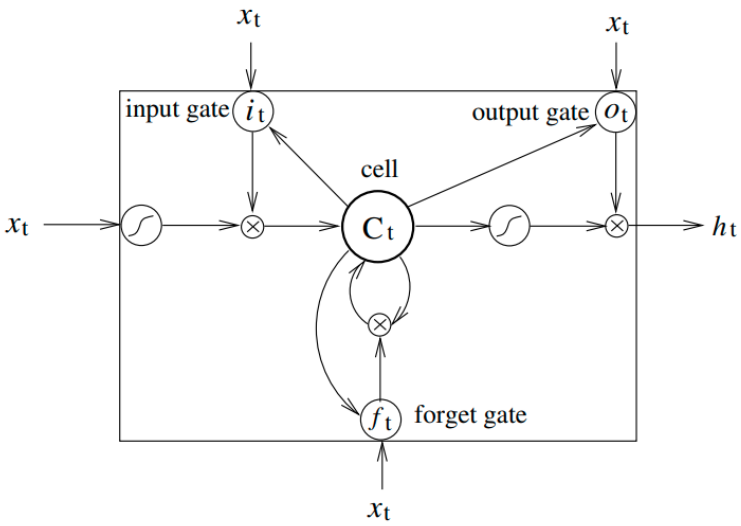

Figura 3.4: Representación de una celda LSTM

En la figura 3.4 puede observarse la representación de una celda LSTM (Huang, Xu, y Yu, 2015), cuya implementación se puede realizar mediante las siguientes funciones:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \tanh c_t
\end{aligned}
$$

siendo σ la función sigmoide logística, y f, i, o y c la puerta del olvido (*forget gate*), la puerta de entrada (*input gate*), la puerta de salida (*output gate*) y el vector de celda (*cell vector*), respectivamente; siendo todos ellos de las mismas dimensiones que el vector oculto h (*hidden vector*). A su vez, W representa la matriz de pesos entre los sufijos, siendo, por ejemplo, W_{hi} la matriz de pesos entre el vector oculto h y la puerta de entrada i.

En la figura 3.5 se representa una red LSTM con un ejemplo del corpus, en la que las celdas LSTM se representan rellenas en gris con líneas.

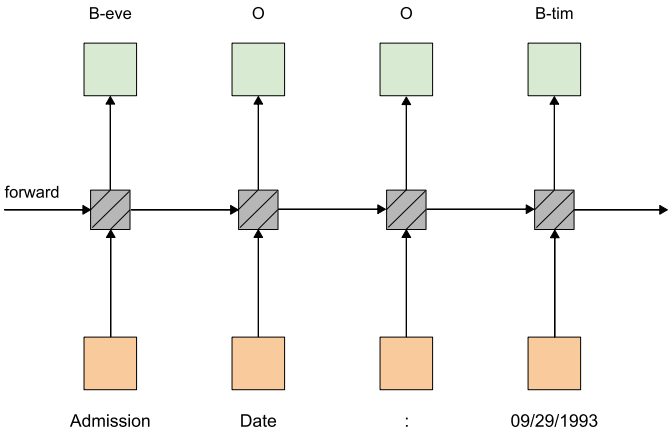

Figura 3.5: Representación de una red LSTM

Dado que en la tarea de etiquetación de secuencias IOB se tiene acceso tanto a los datos pasados como a los futuros, se puede extender la red LSTM de forma que no solo se propague el contexto hacia adelante, sino también hacia atrás, tal y como se propone en (Graves, Mohamed, y Hinton, 2013). Esta red se conoce como LSTM bidireccional o BiLSTM, y queda representada en la figura 3.6.

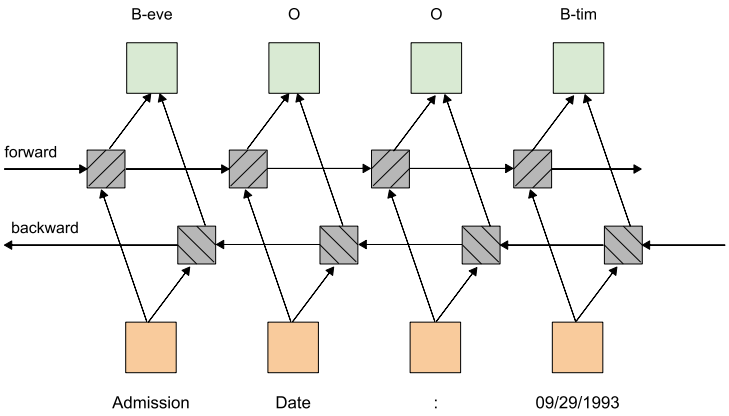

Figura 3.6: Representación de una red LSTM bidireccional

Para el entrenamiento de este modelo, el único rasgo que se utiliza como entrada son los *tokens* que componen el texto. La arquitectura final del sistema, denominado `BiLSTMSR`, queda representada en la figura 3.7. Tras numerosos entrenamientos con esta arquitectura, no se logró conseguir resul-

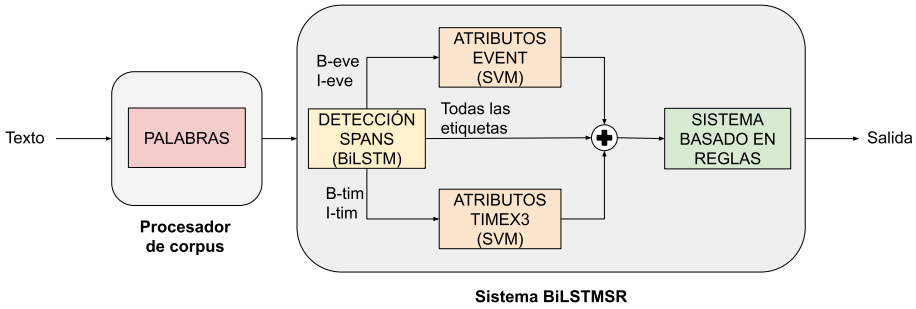

Figura 3.7: Arquitectura del sistema BiLSTMSR

tados satisfactorios, siendo el sistema incapaz de converger con el conjunto de datos de entrenamiento, produciendo resultados de salida casi aleatorios.

Haciendo pruebas con otros corpora de mayor tamaño, del orden de 10 veces más datos para el conjunto de entrenamiento, sí que se conseguían resultados aceptables, por lo que se plantea como hipótesis que el corpus i2b2 es demasiado pequeño para el tipo de sistema. Debido a lo anterior, no se contempla el sistema `BiLSTMSR` en la fase de evaluación.

3.8.2. Detección de *spans* mediante BERT

Debido a los resultados de los experimentos de detección de *spans* con la red BiLSTM, se opta por sustituir este modelo por uno preentrenado, tratando de mitigar los posibles problemas asociados al tamaño del corpus i2b2. Entre todos los modelos disponibles se elige BERT (*Bidirectional Encoder Representations from Transformers*) (Devlin et al., 2019), un modelo basado en Transformers.

Transformer es una red de neuronas basada únicamente en mecanismos de atención, prescindiendo de la recurrencia del planteamiento anterior, así como de las convoluciones propuestas en otros planteamientos, dando lugar a una arquitectura más simple. La arquitectura, descrita originalmente en (Vaswani et al., 2017), queda representada en la figura 3.8.

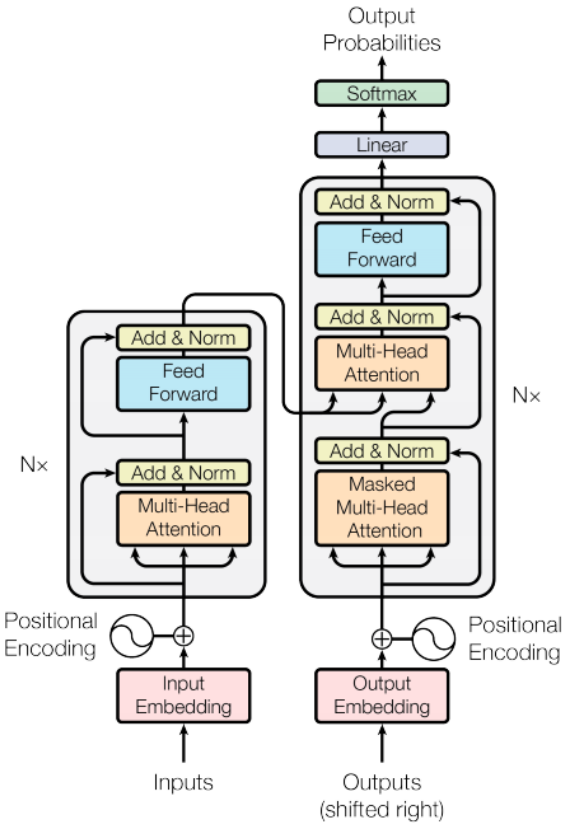

Figura 3.8: Representación de la arquitectura Transformer (Vaswani et al., 2017)

Una función de atención puede describirse como la asociación de una consulta y un conjunto de pares clave-valor con una salida, siendo vectores la consulta (Q), las claves (K), los valores (V) y la salida. La salida se calcula como la suma ponderada de todos los valores, calculándose a su vez los pesos de dichos valores por una función de compatibilidad de la consulta respecto a la clave asociada al valor. La función de atención utilizada se denomina *Scaled Dot-Product Attention*, que toma como entrada las consultas Q y el vector de claves K, ambos de dimensión d_k, junto con los valores V, de dimensión d_v. Dado que se calcula simultáneamente la atención para conjuntos de varias consultas, dando lugar a una matriz, la atención puede calcularse como (Vaswani et al., 2017):

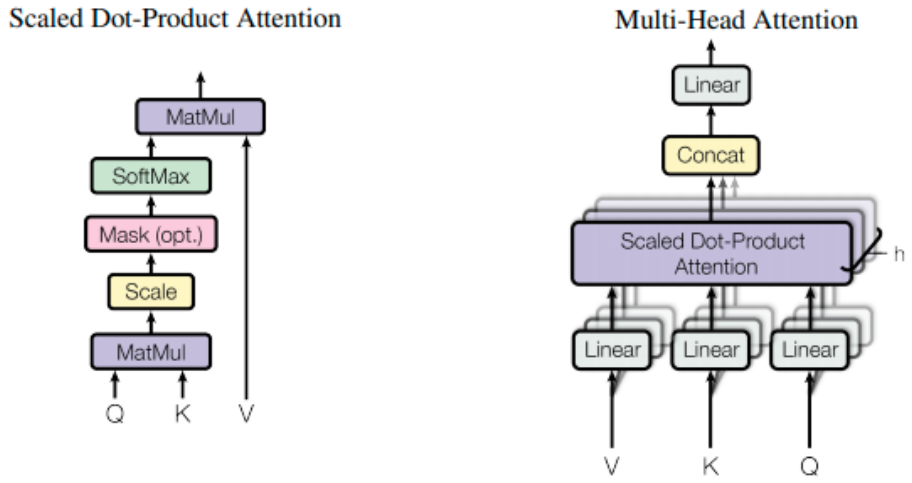

Figura 3.9: Representación de la función de atención *Scaled Dot-Product Attention* (izquierda). Representación de las múltiples cabezas de atención (derecha). (Vaswani et al., 2017)

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

En (Vaswani et al., 2017) se propone el cálculo de la función de atención múltiples veces de forma paralela, dado que beneficia el resultado final. Este cálculo múltiple se denomina *MultiHead*, o de múltiples cabezas de atención, y puede definirse como:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$
$$donde\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

En la fórmula anterior, W_i^Q, W_i^K, W_i^V y W^O son las matrices de parámetros que representan las múltiples ejecuciones paralelas. En la implementación de (Vaswani et al., 2017), que se toma como referencia, se utilizan $h = 8$ capas o cabezas de atención paralelas. Tanto el cálculo en una dimensión de la función de atención, como las múltiples cabezas de atención, quedan representadas en la figura

El modelo tiene una estructura codificador-decodificador de forma que el codificador asocia una secuencia de entrada formada por representacio-

nes simbólicas $(x_1, \ldots, x_n)$ a otra secuencia de representaciones continuas $z = (z_1, \ldots, z_n)$. Dada la secuencia z, el decodificador genera, elemento a elemento, una secuencia de símbolos de salida $(y_1, \ldots, y_m)$. En cada etapa el modelo es autorregresivo, utilizando todos los símbolos generados previamente como datos de entrada adicionales para generar el próximo.

La arquitectura Transformer utiliza capas apiladas y completamente conexas que contienen mecanismos de atención, tanto para el codificador como para el decodificador, representados a la izquierda y la derecha de la figura 3.8, respectivamente. Tanto el codificador como el decodificador se componen por 6 capas apiladas, produciendo una salida de 512 dimensiones.

La arquitectura de BERT está basada en la arquitectura Transformer descrita anteriormente, teniendo una implementación idéntica a la original, variando únicamente en sus dimensiones. El modelo $\text{BERT}_{\text{BASE}}$, utilizado en los experimentos, se compone de $L = 12$ capas apiladas, un tamaño de capa oculta de $H = 768$ unidades y $A = 12$ cabezas de atención, sumando un total de 110 millones de parámetros.

Las representaciones de entrada/salida de BERT están diseñadas para poder gestionar indistintamente secuencias de *tokens* que representen frases y pares de frases, de forma que pueda ser utilizado para una mayor variedad de tareas, como puede ser el Reconocimiento de Entidades Nombradas (*Named Entity Recognition*, NER) o la Búsqueda de Respuestas (*Question Answering*, QA). Dado que la tarea a resolver se engloba dentro de NER, las secuencias de entrada se compondrán de una única frase, por lo que desde este punto se omitirán todos los aspectos relacionados con pares de frases, que pueden ser consultados en (Devlin et al., 2019).

Para generar las representaciones, BERT hace uso de los *embeddings* WordPiece (Wu et al., 2016) con un vocabulario de 30.000 *tokens*. El primer *token* de cada secuencia es siempre el *token* especial de clasificación (`CLS`). El estado final de este *token* se utiliza como representación de la agregación de secuencias en tareas de clasificación, como es el caso de NER. Dado un *token* de la secuencia, su representación de entrada se forma mediante la suma de la representación del propio *token* y los vectores de representación de segmentación y de posición, tal y como se refleja en la figura 3.10 (Devlin et al., 2019).

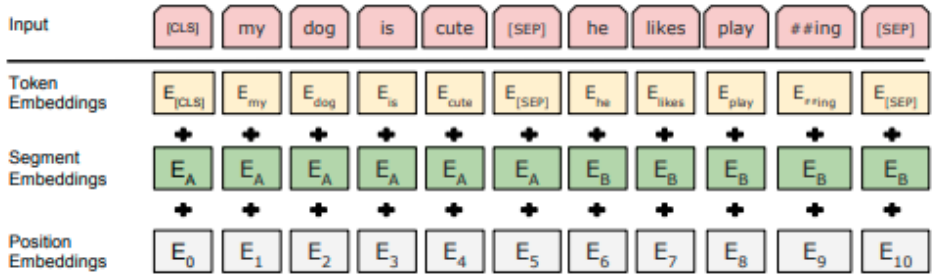

Figura 3.10: Representación de entrada de BERT (Devlin et al., 2019)

Preentrenamiento de BERT

En los experimentos a describir en este documento, se recurre a un modelo de BERT preentrenado. El preentrenamiento del modelo se realiza mediante dos tareas de aprendizaje no supervisado:

1. **LM enmascarado** (*Masked LM*, MLM): con el objetivo de realizar un entrenamiento bidireccional sin recurrir a la concatenación de un modelo entrenado con propagación hacia adelante y otro hacia atrás, se recurre a enmascarar un 15 % de los *tokens* de entrada elegidos de forma aleatoria, con el objetivo de predecirlos. Los vectores finales correspondientes a los *tokens* enmascarados se introducen en una capa `softmax` sobre el vocabulario. Para esta tarea, se utiliza como secuencia de entrada el conjunto de *tokens* de WordPiece al completo. Este proceso permite obtener un modelo preentrenado, pero produce un desajuste entre el proceso de preentrenamiento y el de ajuste (*fine-tuning*), ya que el primero no incluye el *token* `[MASK]` en su vocabulario. Para mitigarlo, (Devlin et al., 2019) proponen no sustituir siempre los *tokens* enmascarados por `[MASK]`, sino que en el entrenamiento se seleccionan un 15 % de las posiciones de los *tokens* de forma aleatoria. En las posiciones elegidas, se sustituye el *token* enmascarado por `[MASK]` un 80 % de las veces, un *token* aleatorio un 10 % de las veces y el propio *token* el 10 % restante.

2. **Predicción de la siguiente frase** (*Next Sentence Prediction*, NSP): con el objetivo de que el modelo sea capaz de capturar relaciones entre frases, se preentrena para la tarea de NSP mediante un corpus monolingüe. Cuando se selecciona cada par de frases `A` y `B` para este proceso, un 50 % de las veces la frase `B` es la frase que sigue a `A`, y el

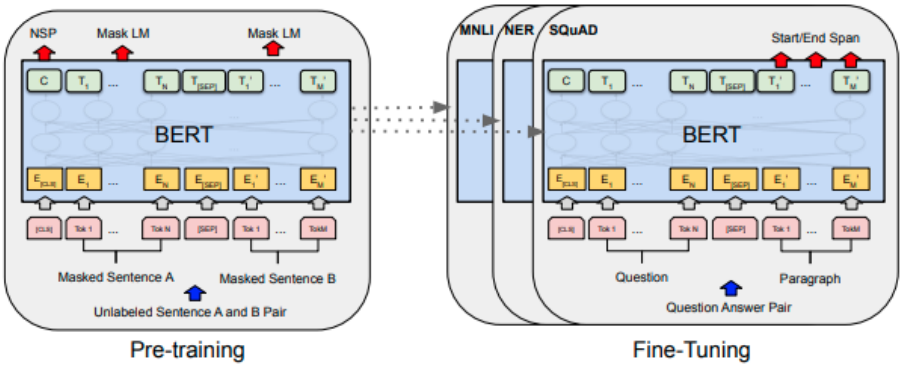

Figura 3.11: Procesos de preentrenamiento y *fine-tuning* de BERT (Devlin et al., 2019)

50 % restante es una frase aleatoria del corpus.

Las tareas anteriores hacen uso de un corpus de entrenamiento compuesto por BooksCorpus (Zhu et al., 2015), formado por 800 millones de palabras, y por textos en inglés extraídos de Wikipedia, conteniendo un total de 2500 millones de palabras. Los textos de Wikipedia se componen únicamente del contenido de las entradas, ignorándose listas, tablas y encabezados.

Fine-tuning de BERT

Partiendo del modelo BERT preentrenado, se realiza un ajuste del mismo para adaptarlo al dominio de la tarea de extracción de eventos y expresiones temporales. Para ello, basta con añadir una capa adicional a BERT, tal y como se ilustra en la figura 3.11 (Devlin et al., 2019), y entrenarlo para etiquetar secuencias del corpus i2b2 del mismo modo que en los sistemas anteriores. La principal ventaja que tiene este modelo respecto al uso de la red BiLSTM es que el modelo preentrenado ya ha utilizado una cantidad ingente de datos, por lo que la etapa de ajuste puede obtener buenos resultados en relativamente poco tiempo, y sin requerir una gran capacidad de cómputo.

En (Devlin et al., 2019) se recomienda realizar el proceso de *fine-tuning* de entre 2 y 4 épocas, aunque también se especifica que el número óptimo de épocas puede variar en función del dominio del problema. Por ello, se realizan pruebas en un margen de 2 a 5 épocas, exponiéndose los resultados en el capítulo 4.

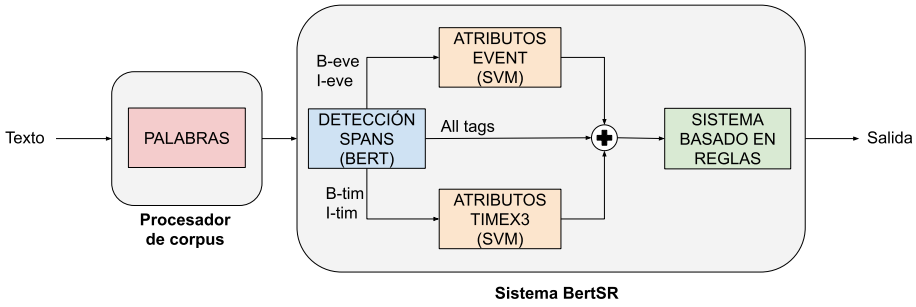

Figura 3.12: Arquitectura del sistema BertSR

El sistema `BertSR` integra el modelo BERT anterior para la detección de *spans*, tomando como entrada el texto preprocesado y etiquetando la secuencia de texto en formato IOB, con los clasificadores de atributos basados en SVM, tal y como se ilustra en la figura 3.12. Dado que este sistema si produce buenos resultados, se incluye entre los sistemas a evaluar.

Capítulo 4

Evaluación

En este capítulo se describe la metodología utilizada para evaluar los diferentes sistemas propuestos en el capítulo 3 para la extracción de eventos y expresiones temporales.

4.1. Metodología de evaluación

Los sistemas propuestos se evalúan mediante una comparativa de las métricas obtenidas mediante el *script* de evaluación del corpus i2b2 (Sun, Rumshisky, y Uzuner, 2013a), que permite calcular la precisión, cobertura (*recall*) y valor F1 para los intervalos de texto (*spans*) de eventos (`EVENT`) y expresiones temporales (`TIMEX3`); así como la exactitud (*accuracy*) para cada uno de los atributos de las entidades: `type`, `polarity` y `modality` para los eventos; y `type`, `val` y `modifier` para las expresiones temporales.

Una vez obtenidas las métricas, se procede a comparar los resultados de los sistemas planteados frente a la clasificación de la competición (Sun, Rumshisky, y Uzuner, 2013b), utilizando el valor F1 para la comparación de detección de *spans* y la exactitud para los atributos de eventos y expresiones temporales.

4.2. Métricas de evaluación

Como se ha expuesto en la sección anterior, las métricas a utilizar para la evaluación de los resultados son la precisión, la cobertura, el valor F1 y la exactitud. La precisión permite cuantificar cuántos elementos se han

clasificado correctamente respecto al total de elementos de salida del sistema, y puede definirse como:

$$precisión = \frac{|\{salida\} \cap \{valor.de.verdad\}|}{|\{salida\}|}$$

donde $\{salida\}$ representa el conjunto de elementos de salida, y $\{valor.de.verdad\}$ el conjunto de valores del *gold standard*. La cobertura, por otra parte, permite cuantificar el número de elementos clasificados correctamente respecto al total de elementos del *gold standard*, pudiendo definirse como:

$$cobertura = \frac{|\{salida\} \cap \{valor.de.verdad\}|}{|\{valor.de.verdad\}|}$$

El valor F1 permite combinar las dos métricas anteriores en un único valor, facilitando la comparativa de los sistemas. Se puede definir como:

$$F1 = 2 \cdot \frac{precisión \times cobertura}{precisión + cobertura}$$

Por último, la exactitud permite medir el porcentaje de elementos correctamente clasificados. Dado que se aplica sobre atributos característicos de cada tipo de entidad, no se compara respecto al total de *tokens* sino respecto al total de entidades correctamente identificadas por su *span*. Puede definirse como:

$$exactitud = \frac{|\{salida.atributos\} \cap \{valor.de.verdad.atributos\}|}{|\{salida.spans\} \cap \{valor.de.verdad.spans\}|}$$

4.3. Resultados

En esta sección se exponen los resultados de los experimentos realizados con los sistemas que se han propuesto en el capítulo 3, resumidos en la tabla 4.1.

Sistema	Descripción
KU1	Replica el trabajo de (Leeuwenberg y Moens, 2017). Sirve como línea base para comparar los sistemas propuestos. Se basa en cuatro SVM, una por cada subtarea.
KU1**	Misma arquitectura que el sistema KU1, pero añade atributos con una ventana de $w = 1$, y aplica la representación unificada de los dígitos de forma discriminada en lugar de a nivel general.
KU2	Amplía la arquitectura del sistema KU1 añadiendo un sistema basado en reglas para el campo value de la expresiones temporales.
KU2_w2	Idéntico al sistema KU2, añadiendo atributos con una ventana de $w = 2$.
CSR	Modificación del sistema KU2_w2, sustituyendo las dos SVM de detección de *spans* por un único CRF, abordando el problema como una clasificación de secuencias.
CSR*	Idéntico al sistema CSR, aplicando regularización L1 y L2 sobre el CRF.
BertSR	Continúa el planteamiento del problema de detección de *spans* como una clasificación de secuencias, sustituyendo el CRF del sistema CSR* por un modelo BERT pre-entrenado.

Tabla 4.1: Descripción de los sistemas evaluados.

4.3.1. Experimento 1: Evaluación previa a la ingeniería de rasgos

En este primer experimento se evalúan todos los sistemas planteados de forma previa a la ingeniería de rasgos, incluyendo, por tanto, todos los sistemas descritos en las secciones 3.3, 3.4 y 3.5, correspondientes a las arquitecturas KU1, KU2, CSR y CSR*. Los resultados de evaluación para EVENTs pueden encontrarse en la tabla 4.2, y los resultados para TIMEX3 en la tabla 4.3.

EVENT							
	Span			Span + atributos			
Sistema	P	R	F	Type acc	Pol acc	Mod acc	Método
KU1	0,83	**0,87**	0,85	0,72	0,83	0,83	SVM +Dígitos unificados
KU1**	0,83	0,86	0,84	0,71	0,82	0,82	SVM
KU2	0,83	0,86	0,84	0,71	0,82	0,82	SVM+RegEx
KU2_w2	0,84	0,86	0,85	0,72	0,83	0,82	SVM+RegEx
CSR	**0,88**	0,86	**0,87**	0,72	0,83	0,83	CRF+SVM+RegEx
CSR*	0,87	**0,87**	**0,87**	**0,73**	**0,85**	**0,84**	CRF(L1 & L2 Reg) +SVM+RegEx

Tabla 4.2: Resultados de la primera evaluación de los sistemas planteados para EVENTs de i2b2.

TIMEX3							
	Span			Span + atributos			
Sistema	**P**	**R**	**F**	**Type acc**	**Val acc**	**Mod acc**	**Método**
KU1	0,85	**0,81**	**0,83**	**0,74**	0	**0,74**	SVM +Dígitos unificados
KU1**	0,85	0,72	0,78	0,64	0,1	0,66	SVM
KU2	0,85	0,72	0,78	0,64	0,4	0,66	SVM+RegEx
KU2_w2	0,86	0,77	0,81	0,67	0,4	0,69	SVM+RegEx
CSR	**0,92**	0,74	0,82	0,64	0,39	0,67	CRF+SVM+RegEx
CSR*	0,89	0,78	**0,83**	0,67	**0,41**	0,7	CRF(L1 & L2 Reg) +SVM+RegEx

Tabla 4.3: Resultados de la primera evaluación de los sistemas planteados para TIMEX3 de i2b2.

En los resultados para EVENTs puede observarse que prescindir de las representaciones unificadas en todos los *tokens* penaliza ligeramente, del orden de 1 centésima, los resultados obtenidos en el sistema KU1** respecto al sistema KU1, a pesar de que también incluye atributos de ventana con $w = 1$. Los resultados se mantienen estables al introducir el sistema basado en reglas en la arquitectura KU2, ya que este afecta únicamente a las expresiones temporales, y muestra una ligera mejora al incluir atributos de ventana con $w = 2$.

Tras plantear la detección de etiquetas IOB como una clasificación de secuencias, en el sistema CSR se observan unas mejoras más notables, de 4 centésimas en el caso de la precisión y 2 en el valor F1, acompañadas de una mejora casi imperceptible en los atributos, con una mejora de 1 centésima únicamente en el atributo modality. Por último, la aplicación de regularización L1 y L2 en el CRF del sistema CSR* permite mejorar entre 1 y 2 centésimas todos los atributos, así como mejora 1 centésima la cobertura en la detección de *spans*, a costa de empeorar una centésima la precisión, manteniéndose el valor F1.

En cuanto a los resultados para TIMEX3, lo primero que destaca es que la exactitud para el atributo value es de 0 para el sistema KU1, debido a la sustitución de todos los dígitos por su representación unificada, a pesar de que obtiene los mejores resultados de cobertura y valor F1 para los *spans*, y exactitud en type y modifier. Este resultado del atributo value motivó la discriminación por el valor del atributo type en el sistema KU1** a la hora de aplicar las representaciones unificadas, de forma que sólo afectase a las

expresiones de tipo *DATE* y *TIME*. Esta modificación permite mejorar el valor de exactitud del atributo `value` en 1 décima, siendo todavía un mal resultado, y a costa de empeorar notablemente el resto de métricas.

El sistema basado en reglas añadido en el sistema `KU2` se incluye para tratar de mitigar los malos resultados para el atributo `value`, diseñando expresiones regulares específicas para recuperar los valores originales de los dígitos en las expresiones temporales a las que afecta la representación unificada de los mismos. Esta medida permite incrementar la exactitud en el atributo en 3 décimas, llegando hasta un valor de 0,4. La inclusión de atributos de ventana en el sistema `KU2_w2` permite mejorar hasta 3 centésimas todas las métricas, a excepción de la exactitud del atributo `value`, que no se ve afectada.

Por último, y al igual que sucedía con los `EVENTs`, el planteamiento de la detección de *spans* como un problema de clasificación de secuencias en el sistema `CSR` permite mejorar notablemente la precisión, aumentando 6 centésimas hasta alcanzar 0,92, aunque a costa de empeorar en 3 centésimas la cobertura, aumentando sólo una centésima el valor F1; y empeorando también los atributos entre 1 y 3 centésimas. La aplicación de regularización L1 y L2 sobre el CRF en el sistema `CSR*` permite equilibrar los valores de precisión y cobertura, aumentando en una centésima el valor F, igualándolo con el resultado conseguido mediante el sistema `KU1`; igualando el resultado de exactitud para el atributo `type` del sistema `KU2_w2`, siendo el segundo mejor; aumentando hasta 0,41 la exactitud del atributo `value`, logrando el mejor resultado; y aumentando hasta 0,7 la exactitud del campo `modifier`, quedando con el segundo mejor resultado tras el sistema `KU1`.

Tras este primer experimento, puede concluirse que el planteamiento de la detección de *spans* como un problema de clasificación de secuencias permite obtener mejores resultados tanto en la detección de `EVENTs` como en la de `TIMEX3`, que la inclusión de rasgos de ventana también permite mejorar los resultados, y que es conveniente aplicar regularización L1 y L2 sobre el CRF a fin de equilibrar precisión y cobertura en la detección de *spans*, mejorando el valor F1, así como el resto de atributos. En este experimento, el sistema que mejores resultados reporta es `CSR*`.

4.3.2. Experimento 2: Evaluación de la ingeniería de rasgos

El segundo experimento consiste en aplicar ingeniería de rasgos para mejorar los resultados de los mejores sistemas evaluados en el experimento anterior (`KU2_w2` y `CSR*`) que hacen uso únicamente de los *tokens* y sus etiquetas POS. Tal y como se expone en la sección 3.6, en primer lugar se valoran rasgos básicos dependientes del texto y, a continuación, se incluyen rasgos específicos del dominio.

Rasgos dependientes del texto

Rasgos	**Span F1**	**Type acc**	**Polarity acc**	**Modality acc**	**Rasgo de ventana**
`EVENT`					
Word, POS tag	0,85	0,72	0,83	0,82	`Sí`
Word.istitle()	0,85	0,73	0,83	0,82	`Sí`
Prefijo & Sufijo, n=2	0,84	0,71	0,82	0,82	`Sí`
Prefijo & Sufijo, n=2*	0,85	0,73	0,83	0,83	`No`
Prefijo & Sufijo, n=3	0,83	0,7	0,82	0,81	`Sí`
Prefijo & Sufijo, n=3*	**0,85**	0,73	**0,83**	**0,83**	`No`
Conflated token	**0,85**	0,73	**0,83**	0,82	`No`
Prefijo & Sufijo, n=4	0,85	**0,74**	0,83	0,82	`No`
Prefijo & Sufijo, n=5	0,85	0,74	0,83	0,82	`No`
Rasgos	**Span F1**	**Type acc**	**Value acc**	**Modifier acc**	**Rasgo de ventana**
`TIMEX3`					
Word, POS tag	0,81	0,67	0,4	0,69	`Sí`
Word.istitle()	0,81	0,69	0,41	0,7	`Sí`
Prefijo & Sufijo, n=2	0,82	0,72	0,45	0,73	`Sí`
Prefijo & Sufijo, n=2*	0,84	0,75	0,47	0,76	`No`
Prefijo & Sufijo, n=3	0,81	0,71	0,44	0,72	`Sí`
Prefijo & Sufijo, n=3*	**0,85**	0,75	0,48	0,76	`No`
Conflated token	**0,85**	**0,76**	**0,49**	**0,77**	`No`
Prefijo & Sufijo, n=4	0,84	0,74	0,47	0,75	`No`
Prefijo & Sufijo, n=5	0,84	0,74	0,47	0,75	`No`

Tabla 4.4: Resultados de ingeniería de rasgos basados en *tokens* sobre el sistema `KU2_w2`. Se muestran de forma incremental, añadiendo cada rasgo a los de filas anteriores, o sustituyendo el de la fila anterior en las marcadas con un asterisco (*).

En las tablas 4.4 y 4.5 se recogen los resultados de aplicar, de forma incremental, los rasgos básicos planteados: el rasgo booleano que determina si el *token* comienza por mayúscula (`word.istitle()`), prefijos y sufijos de diferentes longitudes, y la inclusión de un rasgo adicional, denominado

`conflated token`, que cuente con la representación unificada de los dígitos de cada *token*.

Analizando en primer lugar los resultados del sistema `KU2_w2`, recogidos en la tabla 4.4 se puede observar una ligera mejora, de una centésima en `EVENTs` y dos centésimas en `TIMEX3`, en la exactitud del atributo `type` tras la inclusión del rasgo `word.istitle()`, también contemplado como rasgo de ventana, sin tener impacto en las otras métricas. La adición de prefijos y sufijos de longitud 2 produce que empeoren entre 1 y 2 centésimas todas las métricas de `EVENTs` a excepción del atributo `modality`, al contrario que en las métricas de `TIMEX3`, que mejoran todas entre 1 y 4 centésimas, destacando la mejora del atributo `value` hasta una exactitud de 0,45.

Si los prefijos y sufijos no se incluyen como rasgos de ventana se obtienen resultados aún mejores, recuperando las anteriores pérdidas en las métricas de `EVENTs`, y mejorando en una centésima el rasgo `modality`; así como mejorando entre 2 y 3 centésimas todas las métricas para `TIMEX3`. Sucede lo mismo al incluir los prefijos y sufijos de longitud 3, también como rasgos de ventana, al empeorar las métricas aún más que en el caso de los prefijos y sufijos de longitud 2. Sin añadirlos como rasgos de ventana, se produce una ligera mejora en las métricas de `EVENTs`, imperceptible con la precisión de 2 decimales utilizada en las tablas, así como mejoras de una centésima en las métricas de `TIMEX3`. Debido a estos resultados, no se incluirán los nuevos rasgos como rasgos de ventana.

La inclusión del rasgo `conflated tokens` no produce una mejora perceptible en las métricas de `EVENTs`, empeorando incluso la exactitud del atributo `modality` en una centésima; al contrario de lo que sucede en las métricas de `TIMEX3` mejorando todos los atributos en una centésima, a excepción de la métrica F1 de los *spans*, que se mantiene. Se hacen pruebas también con la inclusión de prefijos de longitud 4 y 5, pero sólo se logra mejorar en una centésima la exactitud del atributo `type` de los `EVENTs`, manteniéndose o empeorando el resto de métricas. Por tanto, se tomará como versión definitiva del sistema `KU2_w2` la que incluye los rasgos `word.istitle()`, los prefijos y sufijos de longitudes 2 y 3, y el rasgo `conflated token`, al conseguir los mejores resultados en todas las métricas de `TIMEX3` y en dos de las cuatro métricas de `EVENTs`.

Los resultados del sistema `CSR*` se obtienen tras haber realizado validación cruzada con el algoritmo K-Fold y $k = 3$, y haber determinado

Rasgos	**Span F1**	**Type acc**	**Polarity acc**	**Modality acc**	**Rasgo de ventana**
`EVENT`					
Word, POS tag	0,87	0,73	0,85	0,84	`Sí`
Word.istitle()	0,87	0,73	0,85	0,84	`Sí`
Word.istitle()*	0,87	0,73	0,85	0,84	`No`
Prefijo & Sufijo, n=2	0,88	0,72	0,84	0,83	`Sí`
Prefijo & Sufijo, n=3	0,87	0,71	0,84	0,83	`Sí`
Prefijo & Sufijo, n=3*	0,88	0,74	0,85	0,84	`No`
Prefijo & Sufijo, n=4	0,88	0,74	0,85	0,84	`No`
Prefijo & Sufijo, n=5	0,88	0,74	0,85	0,84	`No`
Conflated token	**0,89**	**0,74**	**0,85**	**0,84**	`No`
Prefijo & Sufijo, n=6	0,88	0,74	0,85	0,84	`No`
Prefijo & Sufijo, n=7	0,88	0,74	0,85	0,84	`No`
Rasgos	**Span F1**	**Type acc**	**Value acc**	**Modifier acc**	**Rasgo de ventana**
`TIMEX3`					
Word, POS tag	0,83	0,67	0,41	0,7	`Sí`
Word.istitle()	0,83	0,67	0,41	0,7	`Sí`
Word.istitle()*	0,83	0,67	0,41	0,7	`No`
Prefijo & Sufijo, n=2	0,86	0,73	0,47	0,74	`Sí`
Prefijo & Sufijo, n=3	0,86	0,73	0,46	0,74	`Sí`
Prefijo & Sufijo, n=3*	0,87	0,74	0,48	0,75	`No`
Prefijo & Sufijo, n=4	0,87	0,74	0,48	0,75	`No`
Prefijo & Sufijo, n=5	0,88	0,74	0,48	0,76	`No`
Conflated token	**0,88**	**0,74**	**0,48**	**0,76**	`No`
Prefijo & Sufijo, n=6	0,88	0,74	0,48	0,76	`No`
Prefijo & Sufijo, n=7	0,88	0,74	0,48	0,76	`No`

Tabla 4.5: Resultados de ingeniería de rasgos basados en *tokens* sobre el sistema `CSR*`. Se muestran de forma incremental, añadiendo cada rasgo a los de filas anteriores, o sustituyendo el de la fila anterior en las marcadas con un asterisco (*).

que los valores óptimos para los parámetros de regularización L1 y L2 son $c1 = 0{,}2348$ y $c2 = 0{,}0229$, respectivamente. Analizando estos resultados, recogidos en la tabla 4.5, se obtienen conclusiones similares a las anteriores. En este caso, la inclusión del rasgo `word.istitle()` no produce ningún tipo de mejora, permaneciendo sin alterarse las métricas incluso si no se incluye entre los rasgos de ventana. Al añadir prefijos y sufijos de longitud 2, también como rasgos de ventana, produce una mejora de 1 centésima en el valor F1 de la detección de *spans* para `EVENTs`, disminuyendo 1 centésima el resto de métricas; así como una mejora de 3 centésimas en el valor F1 para los *spans* de `TIMEX3`, y mejoras de entre 4 y 6 centésimas en sus atributos.

Si se añaden los prefijos y sufijos de longitud 3, también como rasgos de ventana, empeoran ligeramente el valor F1 de la detección de *spans* y la exactitud del atributo `type` de `EVENTs`, así como la exactitud del atributo `value` de `TIMEX3`, todas ellas en una centésima. Si no se incluyen los prefijos y sufijos como rasgos de ventana, se recupera el valor F1 para los *spans* de `EVENTs` que se había conseguido con los prefijos y sufijos de longitud 2, mejorando la exactitud de sus atributos entre 1 y 2 centésimas; así como todas las métricas de `TIMEX3`, que mejoran en una centésima. Por tanto, al igual que sucedía en el sistema `KU2_w2`, no se contemplarán los prefijos y sufijos como rasgos de ventana y, en este caso, tampoco se contemplará entre ellos el rasgo `word.istitle()`, ya que no produce ninguna mejora apreciable.

Al añadirse los prefijos y sufijos de longitudes 4 y 5 apenas pueden percibirse mejoras debido, a que los resultados de las métricas están expresados con una precisión de 2 decimales, aunque sí llega a apreciarse una mejora de 1 centésima en el valor F1 de los *spans* y la exactitud del atributo `modifier` de `TIMEX3` tras haberse añadido ambas longitudes de prefijos y sufijos. Tras añadirse el rasgo `conflated token` vuelve a producirse una mejora imperceptible en algunas métricas debido a la precisión de 2 decimales de los resultados, a excepción del valor F1 de los *spans* para los `EVENTs`, que mejora en una centésima.

La inclusión de los prefijos y sufijos de longitudes 6 y 7 no permite mejorar ninguna de las métricas, a pesar del aumento de complejidad a la hora de entrenar los modelos, produciendo incluso un ligero empeoramiento a nivel de milésimas de algunas métricas. Por ello, se puede concluir que la versión del sistema `CSR*` con mejores resultados es la que incluye como rasgos `word.istitle()`, los prefijos de longitudes 2 a 5, y el rasgo `conflated token`.

Rasgos específicos del dominio

Tras haber añadido los rasgos dependientes del texto, se añade a los sistemas anteriores el rasgo `UMLSsemtype`, que permite añadir información del dominio del problema a los datos, tal y como se ha expuesto en la sección 3.6.2. La inclusión de este atributo permite mejorar en una centésima el campo `type` de `EVENTs`, tanto en el sistema `KU2_w2` como en el `CSR*`, aumentando respectivamente hasta 0,74 y 0,75. Tras terminar este experimento, quedan

Épocas	**Span F1**	**Type acc**	**Polarity acc**	**Modality acc**
EVENT				
n=2	**0,91**	**0,76**	**0,89**	**0,88**
n=3	**0,91**	**0,76**	**0,89**	**0,88**
n=4	**0,91**	**0,76**	**0,89**	**0,88**
n=5	**0,91**	0,75	0,88	0,87
Épocas	**Span F1**	**Type acc**	**Value acc**	**Modifier acc**
TIMEX3				
n=2	0,9	0,83	**0,53**	**0,82**
n=3	0,9	0,81	0,52	0,81
n=4	**0,91**	**0,83**	0,52	**0,82**
n=5	**0,91**	**0,83**	0,52	**0,82**

Tabla 4.6: Métricas del modelo `BertSR` en función de las épocas del *fine-tuning* de BERT.

establecidas las métricas finales a utilizar para estos sistemas en la comparativa final.

4.3.3. Experimento 3: Evaluación del sistema `BertSR`

Como se ha expuesto en la sección 3.8, en siguiente paso en la fase de experimentación consiste en tratar de mejorar la detección de los *spans*, abordando el problema como una clasificación de secuencias mediante modelos de Deep Learning. Para ello, se propone partir de la arquitectura del sistema `CSR*`, sustituyendo el CRF por el modelo de Deep Learning correspondiente.

Dado que las pruebas con el modelo BiLSTM no han reportado resultados satisfactorios, tal y como se expone en la sección 3.8.1, se opta por no incluirlos en el proceso de evaluación.

Por el contrario, la inclusión del modelo BERT expuesta en la sección 3.8.2 sí permite obtener buenos resultados, dando lugar al modelo `BertSR`. Este modelo utiliza como únicos rasgos de entrada los *tokens* del texto, prescindiendo de las etiquetas POS y los rasgos estudiados en el experimento anterior. Este experimento consistirá, por tanto, en mantener las SVMs de clasificación de atributos previamente entrenadas con el modelo `CSR*`, variando únicamente los parámetros de *fine-tuning* de BERT y evaluando su impacto en las métricas finales.

Dado que únicamente se va a hacer *fine-tuning* del modelo, sólo se estu-

diará el impacto del número de épocas durante las que se ejecuta el proceso. En (Devlin et al., 2019) se recomienda ejecutar durante un rango de 2 a 4 épocas, aunque pueden depender del dominio, por lo que se harán pruebas en el rango de 2 a 5 épocas. Los resultados de estas pruebas quedan reflejados en la tabla 4.6.

En la tabla puede observarse que el número de épocas no tiene una influencia notable en las métricas de `EVENTs` en el rango recomendado, pero comienza a descender la exactitud de los atributos a partir de la 5ª época. En el caso de `TIMEX3` se observan resultados similares, con variaciones de 1 centésima entre los peores valores y los mejores conseguidos. No obstante, cabe destacar que el mejor valor de exactitud para el atributo `value` (0,53) se consigue con tan sólo 2 épocas, reduciéndose en una centésima en el resto de ejecuciones. De la misma forma, se observa que la exactitud de los atributos `type` y `modifier` se reduce en 2 centésimas en la ejecución de 3 épocas, recuperándose los valores anteriores en las ejecuciones con 4 y 5 épocas.

Teniendo en cuenta los resultados anteriores, valorando en conjunto los resultados de `EVENTs` y `TIMEX3`, puede concluirse que el rango de épocas propuesto en (Devlin et al., 2019) se adapta al dominio del problema, y se toma el *fine-tuning* de BERT con 4 épocas como el mejor resultado, estableciéndose estos resultados como los definitivos para ser utilizados en la comparativa.

4.3.4. Comparativa de resultados

Una vez ejecutados todos los experimentos, se puede proceder a comparar sus resultados con los presentados a la competición del corpus i2b2 (Sun, Rumshisky, y Uzuner, 2013b). En la tabla 4.7 quedan reflejados los resultados de los sistemas originales junto con los propuestos en este documento (con los nombres en negrita), ordenados por el valor F1 de los *spans*, y recurriendo sucesivamente a los siguientes atributos en caso igualdad de puntuaciones.

Comparativa de métricas de `EVENTs`

Comenzando por los resultados para `EVENTs`, y partiendo desde las últimas posiciones de la tabla, se encuentra que el sistema `KU1` ocupa la antepenúltima (12ª) posición de la tabla, aventajando en 2 centésimas en el valor F1 a los dos últimos sistemas, y a 7 centésimas de la mejor puntuación. En

Organización /Sistema	**Span F1**	**Type acc**	**Polarity acc**	**Modality acc**	**Método**
`EVENT`					
Beihang University; Microsoft Research Asia; Tsinghua University	**0,92**	**0,86**	0,86	0,86	CRF
BertSR	0,91	0,76	**0,89**	**0,88**	BERT + SVM
Vanderbilt University	0,9	0,84	0,85	0,83	CRF + SVM
The University of Texas	0,89	0,8	0,85	0,84	CRF + SVM
CSR*	0,89	0,75	0,85	0,84	CRF + SVM
The University of Texas-deSouza	0,88	0,71	0,85	0,05	CRF
University of Arizona	0,88	0,73	0,79	0,8	CRF + SVM + NegEx
University of Novi Sad; University of Manchester	0,87	0,82	0,79	0,82	CRF + diccionarios
Siemens Medical Solutions	0,86	0,71	0,78	0,77	CRF + MaxEnt
MAYO Clinic	0,85	0,76	0,75	0,76	CRF
KU2_w2	0,85	0,73	0,83	0,82	SVM
KU1	0,85	0,72	0,83	0,83	SVM
LIMSI-CNRS; INSERM; STL CNRS; LIM&BIO	0,83	0,8	0,84	0,85	CRF + SVM
University of Illinois	0,83	0,74	0,75	0,77	Integer Quadratic Program
Organización /Sistema	**Span F1**	**Type acc**	**Value acc**	**Modifier acc**	**Método**
`TIMEX3`					
Beihang University; Microsoft Research Asia; Tsinghua University	**0,91**	**0,89**	0,72	**0,89**	CRF + SVM + Reglas
BertSR	**0,91**	0,83	0,52	0,82	BERT + SVM + RegEx
MAYO Clinic	0,9	0,86	**0,73**	0,86	RegEx
University of Novi Sad; University of Manchester	0,9	0,85	0,7	0,83	Reglas
The University of Texas	0,89	0,78	0,62	0,79	CRF + SVM + Reglas
Siemens Medical Solutions	0,89	0,86	0,6	0,8	SUTime
The University of Texas-deSouza	0,89	0,78	0,59	0,79	GUTime + CRF + Reglas
University of Arizona	0,88	0,81	0,69	0,8	HeidelTime + CRF
CSR*	0,88	0,74	0,48	0,76	CRF + SVM + RegEx
Vanderbilt University	0,87	0,85	0,7	0,85	Reglas + HeidelTime
KU2_w2	0,85	0,76	0,49	0,77	SVM + RegEx
LIMSI-CNRS; INSERM; STL CNRS; LIM&BIO	0,84	0,75	0,54	0,72	HeidelTime
KU1	0,83	0,74	0	0,74	SVM
Bulgarian Academy of Sciences; American University in Bulgaria; University of Colorado School of Medicine	0,8	0,72	0,61	0,71	RegEx

Tabla 4.7: Resultados de la competición i2b2.

cuanto a los atributos, se encuentra 1 centésima por encima del valor más bajo de `type` y 14 centésimas por debajo de la mejor puntuación; 8 centésimas por encima del peor valor y 6 centésimas por debajo del mejor valor de `polarity`; y 6 centésimas por encima del segundo peor valor y 5 centésimas por debajo del mejor valor de `modality`. Para el último atributo se ha comparado con el segundo peor valor de exactitud (0,77) debido a que el peor (0,05) está muy por debajo del resto, y se considera que no sería

una comparación ilustrativa. Con estos resultados se puede determinar que el sistema `KU1` dista de los mejores resultados en todas las métricas, pero especialmente en la exactitud del atributo `type`.

El sistema `KU2_w2` muestra unos resultados casi idénticos, debido a que la única diferencia que presenta con el sistema a nivel de arquitectura es la inclusión del sistema basado en reglas, pero este afecta únicamente a algunos tipos de `TIMEX3`. Las única métricas que difieren respecto al sistema `KU1` son la exactitud del atributo `type`, que mejora en una centésima, y la del atributo `modality`, que empeora también en una centésima. Debido a estos resultados, aparece en la 11ª posición de la clasificación.

El sistema `CSR*` obtiene resultados notablemente mejores que los anteriores, quedando en la 5ª posición de la clasificación. Supera por 4 centésimas el valor F1 del sistema `KU2_w2`, y queda 3 centésimas por debajo del valor F1 más alto; supera por 2 centésimas la exactitud del sistema `KU2_w2`, y se encuentra 11 centésimas por debajo del valor más alto del atributo `type`; aventaja en 2 centésimas al sistema `KU2_w2` y se encuentra 4 centésimas por debajo del mejor valor de exactitud para el atributo `polarity`; y supera en 1 centésima al sistema `KU1` (el mejor de los dos analizados anteriormente) y queda 4 centésimas por debajo del valor más alto para el atributo `modality`. El sistema presenta unas mejoras discretas frente al anterior sistema propuesto, a excepción del valor F1 para los *spans*, pero sigue encontrándose especialmente distante del mejor valor para el atributo `type`.

Por último, el sistema `BertSR` obtiene una mejora discreta en algunas métricas, pero consigue marcar los mejores resultados en otras, consiguiendo el 2º puesto de la clasificación. Supera por 2 centésimas el valor F1 del sistema `CSR*`, y se encuentra 1 centésima por debajo del mejor valor; aventaja en una centésima al sistema `CSR*` en la exactitud del atributo `type`, quedando 10 centésimas por debajo del mejor valor; consigue el mejor valor para la exactitud del atributo `polarity`, aventajando en 3 centésimas al anterior valor más alto; y también consigue el mejor valor para el atributo `modality`, 2 centésimas por encima del anterior valor más alto. El sistema muestra buenos resultados en términos generales, con una diferencia mínima en la detección de *spans* y logrando la mejor puntuación en dos de los atributos, pero sigue estando muy distante en la métrica del atributo `type`.

Comparativa de métricas de `TIMEX3`

En cuanto a los resultados para las entidades de tipo `TIMEX3`, partiendo nuevamente desde las últimas posiciones de la tabla, se encuentra el sistema `KU1` en la penúltima (13ª) posición, aventajando en 3 centésimas en el valor F1 al peor sistema, y encontrándose 8 centésimas por debajo del mejor resultado. En cuanto a los atributos, se encuentra 2 centésimas por encima del peor sistema y 15 centésimas por debajo del mejor valor del atributo `type`; 3 centésimas por encima del peor sistema y 15 centésimas por debajo del mejor valor del atributo `modifier`, y obtiene el peor resultado para el atributo `value`, obteniendo una puntuación de 0. Con base en los resultados anteriores, puede concluirse que el sistema `KU1` tiene un importante margen de mejora en todas las métricas, especialmente en el atributo `value`.

El sistema `KU2_w2`, planteado como posible solución para mitigar los fallos del sistema `KU1`, incluye un sistema basado en reglas especialmente para mejorar la exactitud del atributo `value`. Este sistema consigue la 11ª posición en la clasificación, aventajando en 2 centésimas a `KU1` en el valor F1 de detección de *spans*, y dista 6 centésimas del mejor sistema; se encuentra 2 centésimas por encima del sistema `KU1` y 13 centésimas por debajo del mejor resultado para el atributo `type`, 3 centésimas por encima del sistema `KU1` y 12 centésimas por debajo del mejor sistema para el atributo `modifier`, y consigue el peor resultado para el atributo `value`, obviando el resultado del sistema `KU1`, con una puntuación de 0,49, 24 centésimas por debajo del mejor resultado. La inclusión del sistema basado en reglas en la arquitectura produce mejoras notables, pero el sistema sigue distando de las puntuaciones más altas en todas las métricas.

El sistema `CSR*`, al contrario que en `EVENTs`, no muestra una mejora sustancial en las métricas respecto al sistema anterior, consiguiendo la 9ª posición de la clasificación. Se encuentra 3 centésimas por encima del sistema `KU2_w2` y 3 centésimas por debajo del valor F1 más alto para la detección de *spans*, 2 centésimas por debajo de la exactitud lograda por `KU2_w2` para el atributo `type`, y 1 centésima por debajo de las puntuaciones de los atributos `value` y `modifier` conseguidas por ese mismo sistema. A excepción de la detección de *spans*, empeoran todas las métricas respecto al anterior sistema, a pesar de que únicamente varía la forma de detectar los *spans*, por lo que, intuitivamente, se esperaba que los resultados se mantuvieran iguales o mejorasen de forma proporcional.

Por último, el sistema `BertSR` sí consigue mejorar todas las métricas, logrando el segundo puesto de la clasificación. Se encuentra 3 centésimas por encima del sistema `CSR*` y empata con la mejor puntuación para el valor F1 en la detección de *spans*, 7 centésimas por encima de `KU2_w2` (el mejor de los planteados hasta el momento) y 6 centésimas por debajo de la mejor puntuación para el atributo `type`, 3 centésimas por encima de `KU2_w2` y 21 centésimas por debajo del mejor resultado para el atributo `value`, y 5 centésimas por encima de `KU2_w2` y 7 centésimas por debajo del mejor sistema para el atributo `modifier`.

Con los anteriores resultados, puede concluirse que el planteamiento de la detección de *spans* como un problema de clasificación de secuencias obtiene mejores resultados que planteándolo como una clasificación estándar, siendo más notable en la detección de `EVENTs` que en `TIMEX3`. En ambos tipos de entidades, los sistemas planteados obtienen resultados relativamente distantes de los mejores para el atributo `type`, lográndose los mejores resultados para los atributos `polarity` y `modality` de `EVENTs`, pero mostrándose también distante de los mejores resultados para los atributos `value` y `modifier` de `TIMEX3`. Es especialmente notable la diferencia de puntuación en el atributo `value`, que se encuentra 2 centésimas por debajo del peor sistema presentado a la competición i2b2.

4.3.5. Significatividad Estadística

Por último, se realizan tests de significatividad estadística para validar la comparativa de los resultados de las arquitecturas propuestas en este trabajo. Se realizan 10 ejecuciones completas (entrenamiento y evaluación) de los sistemas `BertSR`, `CSR*` y `KU2_w2`, a fin de tener muestras que permitan realizar las comparaciones con fiabilidad. No se contempla el sistema `KU1` debido a que es igual que el `KU2_w2`, variando únicamente la inclusión del sistema basado en reglas que afecta exclusivamente a los resultados del atributo `type` de `TIMEX3`.

Para las comparaciones se realizarán dos tipos de test, en función de las muestras: el test-t de Welch (Welch, 1947), que comprueba si las medias de dos muestras son iguales (hipótesis nula) o presentan diferencias (hipótesis alternativa), en caso de que las muestras tengan una distribución normal; y el test de Wilcoxon (Wilcoxon, 1945), que compara los rangos medios de las muestras para determinar si son iguales (hipótesis nula) o presentan

diferencias (hipótesis alternativa), en caso de que las muestras no sigan una distribución normal. La normalidad de las muestras se comprueba mediante el test Shapiro-Wilk (Shapiro y Wilk, 1965).

En la tabla 4.8 se pueden encontrar los resultados de los test de las muestras de los sistemas `CSR*` y `KU2_w2` respecto a las muestras del sistema `BertSR`. En dicha tabla, se acepta la hipótesis nula del respectivo test si el p-valor (p) tiene un valor tal que $p > 0{,}05$, en caso contrario se rechaza, aceptando la hipótesis alternativa. El único par de muestras en el que ambas siguen una distribución normal es el compuesto por los resultados del atributo `type` de `EVENTs` para los sistemas `BertSR` y `CSR*`, por lo que será el único caso en el que se recurra al test de Welch, utilizando el test de Wilcoxon en el resto de muestras.

Sistema	**Span p-valor**	**Type p-valor**	**Polarity p-valor**	**Modality p-valor**
`EVENT`				
CSR*	0,000171	7,036e-12	0,000172	0,000171
KU2_w2	0,000157	0,000162	0,000161	0,000161
Ejecución	**Span p-valor**	**Type p-valor**	**Value p-valor**	**Modifier p-valor**
`TIMEX3`				
CSR*	0,000168	0,000172	0,000171	0,000169
KU2_w2	0,000157	0,000161	0,00016	0,000158

Tabla 4.8: Resultados de los test de significatividad respecto a los resultados del sistema `BertSR`.

Como puede observarse en la tabla 4.8, todos los test muestran una fuerte evidencia a favor de la hipótesis alternativa, determinándose que ni las medias ni los rangos medios de las muestras son iguales en ningún caso, por lo que se puede afirmar que los resultados de un sistema son mayores que los de otro. Comparando los valores obtenidos por las ejecuciones de los sistemas, puede concluirse que los mejores resultados los obtiene el sistema `BertSR`, seguido de `CSR*` y `KU2_w2`, validando los resultados obtenidos en la sección 4.3.4.

Capítulo 5

Conclusiones y trabajo futuro

Este capítulo resume las contribuciones realizadas en este trabajo para la tarea de Extracción de Eventos y Expresiones Temporales en Textos Clínicos. Además, se exponen algunas líneas de trabajo futuro que pueden tomar como base los resultados presentados en este documento.

5.1. Conclusiones

En este trabajo se ha descrito la tarea de Extracción de Eventos y Expresiones Temporales en Textos Clínicos, exponiendo sus orígenes fuera del dominio clínico, en las conferencias TempEval, y su evolución hasta el estado del arte alcanzado en la tarea Clinical TempEval de la conferencia SemEval 2017. Tomando como referencias las mejores arquitecturas presentadas a dicha edición de Clinical TempEval, se han planteado una serie de arquitecturas de sistemas para resolver la tarea, todos con un enfoque de aprendizaje supervisado.

A pesar de que los sistemas tomados como referencia se desarrollaron haciendo uso del corpus THYME, las soluciones presentadas en este documento se han desarrollado sobre el corpus i2b2, debido a que no se disponía de acceso al corpus THYME a fecha de redacción de este documento. No obstante, se toman como referencia los sistemas de Clinical TempEval 2017, debido a que son más recientes y utilizan técnicas más novedosas.

En total se han propuesto 5 arquitecturas para resolver la tarea:

KU1 Una de las arquitecturas presentadas a Clinical TempEval 2017, con el objetivo de medir su efectividad ante un corpus diferente al que se usó para su diseño original, y para establecer unas métricas básicas para cuantificar los efectos de las modificaciones a plantear en los siguientes sistemas. Recurre a una SVM por cada una de las 4 tareas a desempeñar: detección de *spans* de eventos (ES) y expresiones temporales (TS) (ambas mediante etiquetadores en formato IOB), y clasificación de los atributos de eventos (EA) y expresiones temporales (TA). Toma como rasgos las palabras, sus etiquetas gramaticales, y las anteriores como rasgos de ventana, con un tamaño de ventana $w = 1$.

KU2_w2 Basada en la arquitectura KU1, pero añadiendo un sistema basado en reglas para la normalización de valores de fechas (`DATES`) y horas (`TIMES`). Toma los mismo rasgos que el sistema KU1, pero se aumenta el tamaño de ventana a $w = 2$, y añade otros rasgos, como el booleano `word.istitle()` que indica si cada *token* empieza o no por mayúscula, los prefijos y sufijos de longitudes 2 y 3, y la representación unificada de dígitos.

CSR* Partiendo de la arquitectura KU2_w2, se sustituyen las dos SVM encargadas de la detección de *spans* por un único CRF, tratando el texto como una secuencia a etiquetar. Utiliza los mismos rasgos que el sistema KU2_w2, pero incluyendo también los prefijos y sufijos de longitudes 4 y 5. Además, se aplica regularización L1 y L2 sobre el CRF durante el entrenamiento para prevenir el sobreentrenamiento.

BiLSTMSR Idéntica a la arquitectura CSR*, pero sustituyendo el CRF por una red de neuronas LSTM bidireccional, tomando como único rasgo las palabras. Es la única de las arquitecturas que se ha descartado para su evaluación, al no llegar a converger, planteándose como hipótesis la escasez de datos para su entrenamiento.

BertSR Igual que la arquitectura BiLSTMSR, pero sustituyendo la red de neuronas LSTM bidireccional por un modelo BERT pre-entrenado, realizando *fine-tuning* de 4 épocas sobre el corpus i2b2. Al igual que BiLSTMSR, utiliza las palabras como único rasgo. Este sistema supone la principal aportación del trabajo, al no haberse encontrado publicaciones previas que aborden este planteamiento con el corpus i2b2.

Las arquitecturas propuestas parten de uno de los planteamientos de Clinical TempEval 2017, aplicando modificaciones de forma progresiva para incluir tecnologías más avanzadas que permitan igualar o mejorar los mejores resultados del corpus i2b2 con una menor cantidad de procesamiento. El sistema BertSR presentado en este trabajo supone el planteamiento más novedoso, proponiendo una arquitectura nueva que logra buenos resultados recurriendo a las palabras como único rasgo y que, debido a la utilización de un modelo pre-entrenado como BERT, no requiere un entrenamiento costoso ni un conjunto de datos grande, como si ocurre con el sistema BiLSTMSR, cuya red de neuronas BiLSTM bidireccional no puede entrenarse debido a la escasez de datos. Esto hace que la utilización de modelos pre-entrenados sea adecuada para un dominio tan específico como el clínico, donde la disponibilidad de datos es limitada.

El sistema BertSR obtiene los mejores resultados, seguido de CSR*, KU2_w2 y KU1. Realizando una comparativa con los sistemas presentados a la competición del corpus i2b2, se concluye que BertSR obtiene resultados muy cercanos a los del mejor sistema de la competición, igualando o mostrando una diferencia mínima en las tareas de detección de *spans*, e incluso superando los mejores resultados para los atributos `polarity` y `modality` de los eventos. No obstante, se observa un margen de mejora en el atributo `type` de los eventos, y en todos los atributos de las expresiones temporales, destacando la normalización del valor de dichas expresiones, que obtiene una puntuación aproximadamente 2 décimas inferior a la del mejor sistema de la competición.

Por todo lo anterior, puede concluirse que la arquitectura BertSR supone una buena aproximación a la resolución del problema, requiriendo un preprocesado simple de los datos y un tiempo reducido, debido la brevedad del proceso de *fine-tuning*, y permitiendo obtener resultados cercanos al estado del arte del corpus i2b2, aunque mostrando margen de mejora en algunas de las tareas.

5.2. Trabajo futuro

Contextualizando la tarea abordada con el estado del arte, y teniendo en cuenta los resultados obtenidos, pueden determinarse varias líneas de trabajo futuro para extender las soluciones propuestas en este documento.

En primer lugar, puede evaluarse la inclusión de nuevos módulos y rasgos en el sistema BertSR, así como la extensión de las reglas de normalización de fechas y horas, con el objetivo de mitigar las diferencias en las métricas de los atributos en los que el sistema queda en desventaja respecto a otros de la competición i2b2. También resultaría de interés estudiar los resultados del sistema sobre el corpus THYME, permitiendo además evaluar el desempeño de BertSR si se realizan el *fine-tuning* y la propia evaluación sobre documentos de diferentes subdominios clínicos, como son los informes de alta de pacientes con cáncer de colon y de tumores cerebrales.

En segundo lugar, la Extracción de Eventos y Expresiones Temporales suele utilizarse como antecedente para la tarea de Extracción de Relaciones Temporales, permitiendo formar una línea de tiempo sobre los eventos recogidos en un documento, tal y como se ha introducido en el capítulo 2. Por ello, se puede plantear la extensión de las soluciones propuestas en este documento para contemplar la Extracción de Relaciones Temporales.

En el presente trabajo, así como en las anteriores líneas de trabajo futuro propuestas, sólo se contemplan corpus en inglés, por lo que podría resultar de interés la evaluación del sistema en entornos multilingües, evaluando otros idiomas como el español. Dada la escasez de datos del dominio clínico anotados para la tarea de Extracción de Entidades, Expresiones Temporales y Relaciones Temporales, el primer reto a abordar en esta futura línea sería la traducción de corpus ya existentes, o la generación de nuevos corpus en los idiomas de interés.

Bibliografía

Bibliografía

[Barros et al.2016] Barros, Marcia, André Lamúrias, Gonçalo Figueiró, Marta Antunes, Joana Teixeira, Alexandre Pinheiro, y Francisco M Couto. 2016. Ulisboa at semeval-2016 task 12: Extraction of temporal expressions, clinical events and relations using ibent. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1263–1267.

[Bethard et al.2015] Bethard, Steven, Leon Derczynski, Guergana Savova, James Pustejovsky, y Marc Verhagen. 2015. Semeval-2015 task 6: Clinical tempeval. En *proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, páginas 806–814.

[Bethard et al.2016] Bethard, Steven, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, y Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1052–1062.

[Bethard et al.2017] Bethard, Steven, Guergana Savova, Martha Palmer, y James Pustejovsky. 2017. SemEval-2017 task 12: Clinical TempEval. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 565–572, Vancouver, Canada, Agosto. Association for Computational Linguistics.

[Bizer et al.2009] Bizer, Christian, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, y Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3):154–165.

[Bodenreider2004] Bodenreider, Olivier. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

[Boguraev et al.2007] Boguraev, Branimir, James Pustejovsky, Rie Ando, y Marc Verhagen. 2007. Timebank evolution as a community resource for timeml parsing. *Language Resources and Evaluation*, 41(1):91–115.

[Boser, Guyon, y Vapnik1992] Boser, Bernhard E, Isabelle M Guyon, y Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. En *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, páginas 144–152.

[Caselli y Morante2016] Caselli, Tommaso y Roser Morante. 2016. Vuacltl at semeval 2016 task 12: A crf pipeline to clinical tempeval. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1241–1247.

[Chang y Lin2011] Chang, Chih-Chung y Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), Mayo.

[Chikka2016] Chikka, Veera Raghavendra. 2016. Cde-iiith at semeval-2016 task 12: Extraction of temporal information from clinical documents using machine learning techniques. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1237–1240.

[Cohan, Meurer, y Goharian2016] Cohan, Arman, Kevin Meurer, y Nazli Goharian. 2016. Guir at semeval-2016 task 12: Temporal information processing for clinical narratives. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1248–1255.

[Cunningham et al.2011] Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, y Wim Peters. 2011. *Text Processing with GATE (Version 6)*.

[Devlin et al.2019] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, y Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

[Fries2016] Fries, Jason Alan. 2016. Brundlefly at semeval-2016 task 12: Recurrent neural networks vs. joint inference for clinical temporal information extraction. *arXiv preprint arXiv:1606.01433*.

[Graves, Mohamed, y Hinton2013] Graves, Alex, Abdel-rahman Mohamed, y Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. En *2013 IEEE international conference on acoustics, speech and signal processing*, páginas 6645–6649. Ieee.

[Grouin y Moriceau2016] Grouin, Cyril y Véronique Moriceau. 2016. Limsi at semeval-2016 task 12: machine-learning and temporal information to identify clinical events and time expressions. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1225–1230.

[Gupta, Joshi, y Bhattacharyya2015] Gupta, Naman, Aditya Joshi, y Pushpak Bhattacharyya. 2015. A temporal expression recognition system for medical documents by taking help of news domain corpora. En *12th International Conference on Natural Language Processing (ICON)*.

[Hansart et al.2016] Hansart, Charlotte, Damien De Meyere, Patrick Watrin, André Bittar, y Cédrick Fairon. 2016. Cental at semeval-2016 task 12: a linguistically fed crf model for medical and temporal information extraction. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1286–1291.

[Hobbs1993] Hobbs, Jerry R. 1993. The generic information extraction system. En *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

[Hochreiter y Schmidhuber1997] Hochreiter, Sepp y Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Huang et al.2017] Huang, Po-Yu, Hen-Hsen Huang, Yu-Wun Wang, Ching Huang, y Hsin-Hsi Chen. 2017. Ntu-1 at semeval-2017 task 12: detection

and classification of temporal events in clinical data with domain adaptation. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 1010–1013.

[Huang, Xu, y Yu2015] Huang, Zhiheng, Wei Xu, y Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

[Khalifa, Velupillai, y Meystre2016] Khalifa, Abdulrahman, Sumithra Velupillai, y Stephane Meystre. 2016. Utahbmi at semeval-2016 task 12: extracting temporal information from clinical text. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1256–1262.

[Lafferty, McCallum, y Pereira2001] Lafferty, John, Andrew McCallum, y Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[Lamurias et al.2017] Lamurias, Andre, Diana Sousa, Sofia Pereira, Luka A Clarke, y Francisco M Couto. 2017. Ulisboa at semeval-2017 task 12: Extraction and classification of temporal expressions and events. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 1019–1023.

[Lee et al.2016] Lee, Hee-Jin, Hua Xu, Jingqi Wang, Yaoyun Zhang, Sungrim Moon, Jun Xu, y Yonghui Wu. 2016. Uthealth at semeval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1292–1297.

[Leeuwenberg y Moens2016] Leeuwenberg, Artuur y Marie Francine Moens. 2016. Kuleuven-liir at semeval 2016 task 12: Detecting narrative containment in clinical records. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1280–1285.

[Leeuwenberg y Moens2017] Leeuwenberg, Tuur y Marie-Francine Moens. 2017. Kuleuven-liir at semeval-2017 task 12: cross-domain temporal information extraction from clinical records. En *Proceedings of SemEval-2017 International Workshop on Semantic Evaluation*, páginas 1030–1034. ACL.

[Li y Huang2016] Li, Peng y Heng Huang. 2016. Uta dlnlp at semeval-2016 task 12: deep learning based natural language processing system for clinical information identification from clinical notes and pathology reports. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1268–1273.

[Lin et al.2016] Lin, Chen, Dmitriy Dligach, Timothy A Miller, Steven Bethard, y Guergana K Savova. 2016. Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.

[Long et al.2017] Long, Yu, Zhijing Li, Xuan Wang, y Chen Li. 2017. XJNLP at SemEval-2017 task 12: Clinical temporal information ex-traction with a hybrid model. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 1014–1018, Vancouver, Canada, Agosto. Association for Computational Linguistics.

[MacAvaney, Cohan, y Goharian2017] MacAvaney, Sean, Arman Cohan, y Nazli Goharian. 2017. Guir at semeval-2017 task 12: A framework for cross-domain clinical temporal information extraction. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 1024–1029.

[Mani et al.2006] Mani, Inderjeet, Marc Verhagen, Ben Wellner, Chong Min Lee, y James Pustejovsky. 2006. Machine learning of temporal relations. En *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, páginas 753–760, Sydney, Australia, Julio. Association for Computational Linguistics.

[Manning et al.2014] Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, y David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. En *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, páginas 55–60, Baltimore, Maryland, Junio. Association for Computational Linguistics.

[Miller et al.2015] Miller, Timothy, Steven Bethard, Dmitriy Dligach, Chen Lin, y Guergana Savova. 2015. Extracting time expressions from clinical text. En *Proceedings of BioNLP 15*, páginas 81–91.

[Paumier2003] Paumier, Sébastien. 2003. *De la reconnaissance de formes linguistiques à lánalyse syntaxique*. Ph.D. tesis, UPEM.

[Pustejovsky et al.2003a] Pustejovsky, James, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, y Dragomir R Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

[Pustejovsky et al.2003b] Pustejovsky, James, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, y others. 2003b. The timebank corpus. En *Corpus linguistics*, volumen 2003, página 40. Lancaster, UK.

[Pustejovsky et al.2010] Pustejovsky, James, Kiyong Lee, Harry Bunt, y Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. En *LREC*, volumen 10, páginas 394–397.

[Pustejovsky y Verhagen2009] Pustejovsky, James y Marc Verhagen. 2009. Semeval-2010 task 13: evaluating events, time expressions, and temporal relations (tempeval-2). En *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, páginas 112–116.

[Ramshaw y Marcus1995] Ramshaw, Lance A. y Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040.

[Sarath, Manikandan, y Niwa2016] Sarath, PR, R Manikandan, y Yoshiki Niwa. 2016. Hitachi at semeval-2016 task 12: A hybrid approach for temporal information extraction from clinical notes. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1231–1236.

[Sarath, Manikandan, y Niwa2017] Sarath, PR, R Manikandan, y Yoshiki Niwa. 2017. Hitachi at semeval-2017 task 12: system for temporal information extraction from clinical notes. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 1005–1009.

[Saurı et al.2006] Saurı, Roser, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, y James Pustejovsky. 2006. Timeml annotation guidelines version 1.2.1.

[Shapiro y Wilk1965] Shapiro, S. S. y M. B. Wilk. 1965. An analysis of variance test for normality (complete samples)†. *Biometrika*, 52(3-4):591–611, 12.

[Soldaini y Goharian2016] Soldaini, Luca y Nazli Goharian. 2016. Quickumls: a fast, unsupervised approach for medical concept extraction. En *MedIR workshop, sigir*, páginas 1–4.

[Strötgen y Gertz2013] Strötgen, Jannik y Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

[Styler IV et al.2014] Styler IV, William F, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C De Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, y others. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

[Sun, Rumshisky, y Uzuner2013a] Sun, Weiyi, Anna Rumshisky, y Ozlem Uzuner. 2013a. Annotating temporal information in clinical narratives. *Journal of biomedical informatics*, 46:S5–S12.

[Sun, Rumshisky, y Uzuner2013b] Sun, Weiyi, Anna Rumshisky, y Ozlem Uzuner. 2013b. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

[Tissot et al.2015] Tissot, Hegler, Genevieve Gorrell, Angus Roberts, Leon Derczynski, y Marcos Didonet Del Fabro. 2015. UFPRSheffield: Contrasting rule-based and support vector machine approaches to time expression identification in clinical TempEval. En *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, páginas 835–839, Denver, Colorado, Junio. Association for Computational Linguistics.

[Tourille et al.2016] Tourille, Julien, Olivier Ferret, Aurélie Névéol, y Xavier Tannier. 2016. Limsi-cot at semeval-2016 task 12: Temporal relation

identification using a pipeline of classifiers. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, páginas 1136–1142.

[Tourille et al.2017] Tourille, Julien, Olivier Ferret, Xavier Tannier, y Aurélie Névéol. 2017. Limsi-cot at semeval-2017 task 12: Neural architecture for temporal information extraction from clinical narratives. En *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, páginas 597–602.

[Turmo2004] Turmo, Jordi. 2004. Information extraction, multilinguality and portability. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 8(22):57–78.

[Uzuner et al.2011] Uzuner, Özlem, Brett R South, Shuying Shen, y Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

[UzZaman et al.2013] UzZaman, Naushad, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, y James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. En *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, páginas 1–9.

[Vaswani et al.2017] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, y Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

[Velupillai et al.2015] Velupillai, Sumithra, Danielle L Mowery, Samir Abdelrahman, Lee Christensen, y Wendy Chapman. 2015. Blulab: Temporal information extraction for the 2015 clinical tempeval challenge. En *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, páginas 815–819.

[Verhagen et al.2007] Verhagen, Marc, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, y James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. En *Proceedings of*

the 4th International Workshop on Semantic Evaluations, SemEval '07, página 75–80, USA. Association for Computational Linguistics.

[Welch1947] Welch, B. L. 1947. The generalization of "Student's"problem when several different population variances are involved. *Biometrika*, 34(1-2):28–35, 01.

[Wilcoxon1945] Wilcoxon, Frank. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

[Wu et al.2016] Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, y others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Xu et al.2012] Xu, Yan, Kai Hong, Junichi Tsujii, y Eric I-Chao Chang. 2012. Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. *Journal of the American Medical Informatics Association*, 19(5):824–832.

[Xu et al.2013] Xu, Yan, Yining Wang, Tianren Liu, Junichi Tsujii, y Eric I-Chao Chang. 2013. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):849–858.

[Zhu et al.2015] Zhu, Yukun, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, y Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.