Universidad de La Habana
Facultad de Matemática y Computación

Metodología para el Reconocimiento de Entidades Nombradas en mensajes cortos

Autor: **Laila González Fernández**

Tutores: **Dr. Yudivián Almeida Cruz**
MSc. Suilán Estévez Velarde

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Junio de 2017

A mis padres, por su dedicación y amor a pesar de la distancia.
A mi hermana, por ser mi mejor confidente y amiga.
A mis abuelos, por su ejemplo.
A Mario, por hacerme feliz y ser mi constante.
A todos mis maestros, por inspirarme y encender
mi curiosidad durante mis años de estudio.

Agradecimientos

A mi familia, en especial a mis padres, por estar siempre cerca de mí, por sus consejos, por llevarme con ellos a conocer el mundo y dedicar su vida a la felicidad de nuestra familia. Estoy muy feliz de poderlos hacer sentir orgullosos con esta tesis. A mi hermana, por ser mi mejor compañía, mi personita preferida en el mundo, por ser tan dulce y hacer tantas veces de hermana mayor. A mis cuatro abuelos que son mis ejemplos a seguir. A mis tíos: Raisa, Olgui y Víctor, por llevarme en sus aventuras. A mis primos: Iván, May, Thali y Vitico, por ser como mis hermanos.

A Mario, por hacer suyos mis problemas, por compartir mis metas, planes y sueños, y hacerme infinitamente feliz. A toda su familia, por todo el cariño que me han dado.

A todos mis amigos y amigas de esa maravillosa etapa de mi vida que fue el preuniversitario. A la profe Geidys, que fue fundamental en mi decisión de estudiar esta carrera que tanto amo; y a la profe Vilma, por quien me habría gustado estudiar Letras también.

A todos mis compañeros en estos cinco años de carrera; en especial, a los que nos ha tocado compartir madrugadas de estudio y me han elegido para compartir sus días de diversión: Mario, Carlos, Jose, Amalia, Kike, JJ y Juan Alberto.

A todos mis profesores, en especial a los del Departamento de Redes por todas las enseñanzas y a Idania por ser ejemplo para mí en todo. A Ale, Gilberto y Darío por hacer que mi última etapa en la Universidad haya sido tan divertida y provechosa.

Un agradecimiento especial para mis tutores, Yudivián y Suilán, por adentrarme en el mundo de la Minería de Textos que tanto me apasiona, por su motivación, sus orientaciones y la confianza que depositaron en mí.

A todos los que hicieron esto posible: un millón de gracias.

Opinión del tutor

Twitter es una red social que ha devenido en un medio de comunicación global. En ella generan y consumen informaciones personas e instituciones que pueden ser desde un estudiante hasta un presidente. Estas características han convertido a *Twitter* en un escenario relevante para el análisis de información.

Los elementos distintivos de los *Tweets* -su brevedad, sus etiquetas propias y las dinámicas generadas en su uso- han hecho que las problemáticas propias del análisis de información tengan otras dimensiones de solución. Es así que problemas como la detección de idiomas, la minería de opinión, la detección de tópicos, entre otros, se han retomado desde diferentes aristas para su aplicación en *Twitter*. Dentro de este grupo de problemas, el Problema de Reconocimiento de Entidades, ha sido también de mucho interés.

Es justamente este problema el que aborda la estudiante Laila González Fernández en su tesis de licenciatura titulada "Metodología para el Reconocimiento de Entidades Nombradas en mensajes cortos". En esta investigación, la estudiante enfrenta el reto de reconocer entidades nombradas en textos caracterizados por su brevedad, el alto nivel de ruido y con gran escasez de contexto. Para ello, basa su hipótesis en la utilización de algoritmos de aprendizaje semi-supervisado y *wikis* como bases de conocimiento externas. Además, evalúa la utilización de las mayúsculas como elemento discriminante para determinar las distintas entidades.

Para realizar el trabajo tuvo que realizar un amplio estudio de la literatura donde fue capaz de analizar y asimilar trabajos en dominios de conocimiento nuevos para la diplomante. Además, utilizó y comparó distintos algoritmos bases para la metodología que se propone con el fin de identificar a aquellos que mejores resultados ofrecen.

Finalmente, para validar su propuesta, realizó un conjunto de experimentos que le permitieron verificar la certeza de la misma. Además, en estos experimentos realizó comparaciones con las otras propuestas reportadas en la literatura y sus resultados fueron superiores.

Todo este trabajo, y hay que destacarlo notablemente, Laila lo realizó con total autonomía e independencia mostrando lo que ya sabía, que es una de las mejores estudiantes de su año. El documento, correctamente presentado, es un ejemplo más que se suma a su excelente desempeño estudiantil.

Por estas razones pedimos al tribunal que otorque a la estudiante Laila González Fernández la máxima calificación posible como colofón a una excepcional trayectoria estudiantil.

Resumen

Los mensajes cortos (como *tweets* y SMS) son una potencial fuente de datos actualizados continua e instantáneamente. La escasez de contexto y la informalidad de estos mensajes constituyen un reto para los sistemas tradicionales de Reconocimiento de Entidades Nombradas. La mayor parte de los esfuerzos realizados en este sentido se basan en técnicas de aprendizaje de máquinas supervisado. Estas técnicas resultan costosas en términos de la recolección de datos y el tiempo de entrenamiento. En esta tesis se presenta un enfoque semi-supervisado para el Reconocimiento de Entidades utilizando *self-training*. Se utilizan *wikis* como bases de conocimiento externas y características no supervisadas para mejorar la portabilidad del sistema. Se evalúa, además, para cada mensaje si el uso de mayúsculas es el adecuado, para evitar uno de los problemas más comunes de los sistemas tradicionales de reconocimiento de entidades al enfrentarse a ambientes como *Twitter*: una excesiva dependencia en la letra inicial mayúscula como indicador de la presencia de una entidad. Al aplicar esta metodología se obtienen resultados comparables con los sistemas de reconocimiento de entidades más utilizados en la actualidad y con mejor rendimiento reportado en la literatura. Los resultados obtenidos validan la efectividad de la metodología desarrollada.

Abstract

Short messages (like tweets and SMS) are a potentially rich source of continuously and instantly updated information. The lack of context and the informality of such messages are challenges for traditional Named Entity Recognition systems. Most efforts done in this direction rely on supervised machine learning techniques which are expensive in terms of data collection and training. In this thesis we present a semi-supervised approach to Named Entity Recogniction using self-training. We use wikis as external knowledge and unsupervised features to improve portability. Whether or not the use of case in a *tweet* is correct is also evaluated. This avoids one of the most common problems of traditional named entity recognition systems when facing noisy environments like *Twitter*: an excessive dependence on title case as an indicator of the presence of an entity. The results obtained when applying this methodology ara similar to those achieved by the most popular and efficient named entity recognition systems. These results validate the effectiveness of the methodology.

Índice general

Índice de figuras

Índice de tablas

Introducción

Internet es una plataforma para el intercambio de información con un volumen de datos y un número de usuarios cada vez mayor. Su creciente nivel de accesibilidad lo ha convertido en una fuente de contenidos de la más diversa índole.

La aparición de la Web 2.0 a mediados de la década pasada ha facilitado aún más el intercambio en la red, convirtiendo a cada uno de sus usuarios en potenciales generadores de información. Estas características de la Web la han colocado en la mira tanto de usuarios y consumidores de servicios e información, como de empresas, partidos políticos y otros organismos dependientes de la opinión pública. Muchas de estas entidades buscan en la web información para apoyar sus procesos de toma de decisiones.

Con este fin, cobran especial relevancia las redes sociales de *microblogging* que permiten a sus usuarios publicar mensajes cortos que generalmente constan solo de texto. El atractivo de estas redes sociales se debe principalmente a la inmediatez que ofrecen, al gran número de mensajes disponibles para analizar y a la variedad de temas que abordan sus usuarios.

Motivación

La más popular de las redes sociales de *microblogging* es Twitter que cuenta ya con 317 millones de usuarios activos y tiene un flujo de 6000 mensajes por segundo[50]. La variedad e inmediatez del contenido presente en las redes sociales hace que resulte atractivo realizar tareas de extracción de información y procesamiento de lenguaje natural en este dominio. Esto permite recopilar y organizar la información e identificar los tópicos más relevantes, los usuarios más influyentes o la opinión de los usuarios respecto a un tema. Para esto es un paso importante el Reconocimiento de Entidades Nombradas.

El Reconocimiento de Entidades Nombradas (NER por sus siglas en in-

glés) busca reconocer en un texto menciones a elementos pertenecientes a ciertas categorías. Estas categorías pueden o no estar predefinidas y generalmente incluyen nombres de personas, organizaciones, lugares y productos. El Reconocimiento de Entidades es imprescindible para muchas otras tareas de la extracción de información entre las cuales se encuentran el enlace de entidades nombradas, la resolución de correferencias y la extracción de relaciones. También, contribuye a mejorar los resultados de otras tareas como la minería de opinión, la búsqueda automática de respuestas y la categorización de textos[1]. Este proceso es parte importante de diversas aplicaciones del análisis de redes sociales como el estudio del impacto de un producto y sus competidores en el mercado[70], el análisis de debates electorales e intención de voto[64][97] y el diseño de sistemas para dar respuestas rápidas en casos de desastres[54][72].

Los sistemas empleados en la actualidad para el Reconocimiento de Entidades en textos largos, estructurados y en idioma inglés alcanzan una precisión superior al 90 %. Los resultados de estos algoritmos, al ser usados en el análisis de mensajes cortos presentes en redes sociales, son significativamente inferiores[84].

Antecedentes

El grupo de Inteligencia Artificial de la Universidad de La Habana tiene como una de sus líneas de trabajo el análisis de redes sociales, con especial énfasis en *Twitter*. En este sentido se han desarrollado investigaciones relacionadas con la detección de idioma[3], la detección de tópico[67], análisis de influencias[48] y la minería de opinión[33]. Es un objetivo del grupo de investigación la creación de una plataforma completa de análisis de *Twitter* donde se integrarán todas estas investigaciones. La metodología para el Reconocimiento de Entidades Nombradas en *Twitter* que se presenta en este trabajo será otro de los elementos importantes de ese sistema.

Problemática

Las personas, lugares, organizaciones y otras entidades que se mencionan en un texto son algunos de los aspectos más relevantes que pueden ser extraídos del mismo. Para un ser humano puede resultar sencillo identificar y clasificar entidades nombradas en ciertos dominios. Sin embargo, hacer esto de forma automática constituye una tarea computacionalmente compleja. La dificultad del proceso es aún mayor en textos cortos con escasez

de contexto y mucho ruido. Los pocos sistemas diseñados con el objetivo de reconocer entidades en este tipo de mensajes son muy dependientes del dominio y requieren un gran volumen de datos anotados manualmente.

Hipótesis

El desarrollo de una metodología diseñada específicamente para el Reconocimiento de Entidades Nombradas en mensajes cortos para el idioma español permitirá mejorar los resultados obtenidos al emplear sistemas de propósito general o enfocados en el análisis de textos estructurados. El uso de diversas *wikis* como base de conocimiento unido al uso de características no supervisadas hará posible desarrollar un sistema que sea independiente del dominio. El uso de aprendizaje semi-supervisado eliminará la necesidad de contar con un gran corpus anotado para el Reconocimiento de Entidades.

Objetivo

Objetivo General

- Diseñar una metodología para el Reconocimiento de Entidades capaz de adaptarse a nuevos tipos de entidades en mensajes cortos, dominio donde habitualmente no existen corpus anotados o no son de libre acceso.

Objetivos Específicos

- Estudiar el estado del arte concerniente al Reconocimiento de Entidades Nombradas.
- Identificar las características del Reconocimiento de Entidades Nombradas en mensajes cortos.
- Desarrollar una metodología que emplee aprendizaje semi-supervisado atendiendo a la escasez de corpus anotados en diversos dominios.
- Desarrollar una metodología capaz de emplear diversas bases de conocimiento según el dominio en estudio.
- Evaluar la eficacia de la metodología en Twitter y otros dominios.

Estructura de la Tesis

El contenido de la tesis se encuentra dividido en tres capítulos. En el primer capítulo se presenta una definición formal del problema de Reconocimiento de Entidades Nombradas y se analizan los enfoques existentes para darle solución. En el segundo capítulo se introduce la metodología propuesta. Además, se exponen los tres conjuntos de características que se utilizan en la metodología: clásicas, no supervisadas y globales; y los dos tipos de clasificadores que se usan durante el entrenamiento: clásicos y estructurados. En el tercer capítulo se analizan los resultados obtenidos utilizando distintos conjuntos de características y clasificadores. También, se compara la propuesta final con dos de los sistemas de reconocimiento de entidades más utilizados y eficientes según la literatura y se presentan los resultados de la metodología en otro dominio.

Al final de la tesis se presentan las conclusiones obtenidas tras la investigación y se listan recomendaciones para futuros trabajos. Se encuentran, asimismo, la bibliografía y anexos que complementan el trabajo.

Capítulo 1

Reconocimiento de Entidades Nombradas

La Minería de Textos es la tarea de analizar documentos procedentes de diversas fuentes con el objetivo de descubrir información y conocimiento que anteriormente no se conocía[92]. Surge en la década de 1980 como una tarea manual. La creciente cantidad de documentos disponibles en forma digital y la necesidad de organizarlos y aprovechar el conocimiento contenido en ellos ha provocado un auge de los sistemas automatizados de extracción de información. La minería de textos es, actualmente, un campo con un gran valor comercial que se nutre de otras áreas como la recuperación de información, la minería de datos, el aprendizaje automático, la estadística y el procesamiento de lenguaje natural.

Una de las tareas más estudiadas de la Minería de Textos es el Reconocimiento de Entidades Nombradas. Este es un campo con más de veinte años de estudio. Sus objetivos son la extracción y clasificación de entidades nombradas en textos. Por entidades nombradas se entiende las menciones a designadores rígidos[71]. Un designador rígido es definido a su vez por la lógica modal y la filosofía del lenguaje como una expresión que se refiere a una misma entidad en todos los contextos posibles en los que esa entidad existe, y no designa nada en aquellos contextos donde no existe[57]. Ejemplos de designadores rígidos son: los nombres propios, especies biológicas y expresiones temporales y numéricas.

Esta tarea se abordó por primera vez en 1996[41]. Desde ese momento se apreció la importancia de reconocer unidades de información como nombres de personas, lugares y organizaciones; y expresiones numéricas como horas, fechas y precios. El Reconocimiento de Entidades pasó entonces a ser una

de las principales componentes de la extracción de información[71].

Los tres tipos de entidades más estudiados son los nombres de personas, lugares y organizaciones[6]. Los trabajos publicados suelen incluir, además, la categoría «misceláneas». Otras categorías frecuentes son: producto, evento y las categorías numéricas referidas anteriormente. Trabajos más recientes no limitan los posibles tipos de entidades a extraer[2]. Otros emplean jerarquías de entidades e incluyen más de cien categorías[89] como color, animal y religión.

El Reconocimiento de Entidades desempeña un papel muy importante en otros problemas relacionados con la minería de texto, tales como la búsqueda automática de respuestas y la categorización de textos. Se ha utilizado en problemas de atención a clientes, seguimiento de noticias, limpieza y preparación de datos, motores de búsqueda y monitoreo de tendencias[87][71]. Representa también un enorme avance en la biología y la genética, pues permite a los investigadores buscar en la abundante literatura menciones a genes y proteínas[55][46][61].

La mayor parte de los estudios se centran en el idioma inglés[43], aunque desde la Conferencia de Aprendizaje Computacional del Lenguaje Natural (CoNLL 2003)[86] han aparecido más trabajos centrados en textos en español. También hay enfoques que intentan reconocer entidades independientemente del idioma del texto.

Aunque se han hecho trabajos en muchos dominios, portar un sistema a un nuevo dominio o género de texto continúa siendo un reto. Técnicas con buenos resultados en un dominio o incluso idioma no necesariamente se trasladan bien a otros[71]. Al probar los sistemas desarrollados durante el torneo MUC-6[41] para ser utilizados en cables de noticias, en un corpus compuesto por correos electrónicos y transcripciones de conversaciones telefónicas, se reportan caídas de entre un 20 % y un 40 % en su precisión y exhaustividad[78].

1.1. Definición del Problema de Reconocimiento de Entidades

Un sistema de reconocimiento de entidades debe delimitar y clasificar menciones a entidades. Dada una secuencia de componentes léxicas o *tokens*, debe decidir si cada *token* es el inicio de una entidad, está dentro de una entidad o está fuera de cualquier entidad.

Uno de los estándares empleados para anotar entidades nombradas en un texto es el formato *BIO*[86]. Este sistema utiliza clases de la forma $B-X$,

$I-X$ y O, donde X es el nombre de una categoría. Un *token* marcado con $B-X$ es el primer *token* de una entidad de tipo X. Uno marcado como $I-X$ pertenece a una entidad de tipo X, pero no es su primer *token*. Uno marcado con O no pertenece a ninguna entidad. El sistema requiere $2|C|+1$ clases, donde $|C|$ es el número de categorías de entidades a identificar.

El uso de este formato para la anotación se ilustra en la figura 1.2. En este ejemplo se encuentran anotadas entidades de dos tipos: organización (ORG) y lugar (LOC). No se permiten anotaciones que se superpongan; esto provocaría que un *token* tuviera dos anotaciones distintas.

Figura 1.1: Tweet tomado de @OnCuba

Danza Contemporánea de #Cuba fascina en #ReinoUnido.

B-ORG I-ORG I-ORG I-ORG O O B-LOC

Figura 1.2: Ejemplo de anotación en un *tweet*

1.2. Reconocimiento de Entidades en Twitter

El Reconocimiento de Entidades Nombradas tiene una precisión reportada de alrededor del 90 %[36][80]. Sin embargo, existen pocos trabajos do-

cumentados sobre la construcción de una herramienta de extracción de información en *tweets* u otros textos de estructura similar. La mayor parte de los trabajos existentes en el campo están diseñados para la extracción de entidades en corpus de noticias y otros textos largos y revisados. La precisión obtenida por estos trabajos disminuye drásticamente al ser empleados para detectar entidades en Twitter[63][84].

El Reconocimiento de Entidades en *tweets* es una nueva y desafiante área de investigación. Esto se debe a que dichos mensajes muchas veces contienen abreviaturas, *hashtags*, emoticonos y errores ortográficos. Asimismo, presentan un uso inadecuado de mayúsculas y minúsculas en las que se apoyan mucho los sistemas tradicionales de extracción de entidades[38][30]. Otro problema es el hecho de que las entidades que aparecen en estos escenarios son distintas de las que aparecen en otros tipos de textos[84].

Por otra parte, debido a la volubilidad de los temas que se tratan en mensajes generados por usuarios, un sistema de extracción de entidades para este dominio debe ser capaz de encontrar entidades que no han sido catalogadas con anterioridad. Constituye otro obstáculo la escasez de contexto que presentan estos mensajes, lo cual dificulta la desambiguación de entidades[84]. Otro reto es el hecho de que los mensajes publicados en redes sociales contienen una variedad de estilos no presente en otros contenidos. Los cambios en los tópicos abordados en las redes sociales son más frecuentes y de una mayor magnitud que en otros medios[28]. Finalmente, es una dificultad la inexistencia de un corpus de magnitud significativa de mensajes cortos con sus entidades nombradas identificadas y clasificadas.

Existen pocos sistemas de reconocimiento de entidades diseñados para ser usados en *Twitter*. Muchos investigadores emplean sistemas diseñados para uso en textos más formales o entrenan estos sistemas para adaptarlos el estilo informal de los *tweets*[44]. Entrenar estos sistemas requiere la anotación de un gran número de *tweets* lo cual es una tarea costosa.

En la literatura se identifican *NERD-ML*[32] y *T-NER*[84] como las principales herramientas existentes diseñadas para reconocer entidades en *Twitter*[29]. Las dos herramientas están entrenadas para reconocer entidades en idioma inglés y no permiten realizar modificaciones con facilidad[29].

T-NER divide el Reconocimiento de Entidades en dos etapas: la segmentación de las entidades (*T-SEG*) y su clasificación (*T-CLASS*). *T-SEG* requiere un corpus anotado y utiliza campos aleatorios condicionales para aprender e inferir reglas para la segmentación de entidades. *T-CLASS* utiliza aprendizaje semi-supervisado para clasificar las entidades usando *LabeledLDA*[79] para hallar similitudes entre el contexto de la entidad y una base de conocimiento.

NERD-ML recopila información de diversas bases de conocimiento y utiliza las salidas de dos sistemas de reconocimiento de entidades: *Stanford NER*[42] y *T-NER* como entrada para un algoritmo de aprendizaje supervisado.

1.3. Enfoques empleados en el Reconocimiento de Entidades

Las propuestas computacionales para la identificación automática de entidades abordan un gran número de estrategias que pueden dividirse en tres grandes conjuntos: basados en reglas, en bases de conocimiento y en aprendizaje de máquinas.

1.3.1. Sistemas basados en reglas

Los primeros sistemas de reconocimiento de entidades datan de la década de 1990 y emplean reglas manuales y heurísticas para detectar entidades[71]. Este enfoque busca pistas en la estructura y la gramática del texto que indican al sistema la presencia de una entidad nombrada. Una de las componentes claves de estos sistemas son las expresiones regulares que permiten identificar fechas, números de carnet de identidad y precios. Estos sistemas también emplean heurísticas derivadas de la morfología y la semántica de la secuencia de entrada. Tienen la ventaja de ser computacionalmente sencillos, pero tienen una baja exhaustividad[6].

En la actualidad, este enfoque suele utilizarse en combinación con otras técnicas[71][21]. Pueden ser usados como atributos en sistemas basados en aprendizaje de máquinas o para identificar candidatos a entidades en una base de conocimientos.

Las reglas más frecuentes se basan en el uso de mayúsculas y en la aparición de *tokens* que hacen función de *triggers*. Este ese el caso de Sr., Dr. y M.Sc. que suelen anteceder una entidad nombrada de tipo persona y de Inc. y Co. que suelen aparecer después de entidades de tipo organización. Asimismo, se buscan preposiciones y otras categorías gramaticales que puedan indicar la presencia de una entidad.

1.3.2. Sistemas basados en bases de conocimiento

Algunas entidades pueden ser reconocidas y clasificadas fácilmente utilizando bases de conocimiento. Un ejemplo sencillo de su uso es la consulta de listas de nombres comunes o países para identificar personas o lugares.

Este enfoque requiere la elaboración manual de una base de conocimiento o un enfoque dinámico que permita su extracción partiendo de un corpus u otra fuente externa. Los trabajos en esta área han estado encaminados al desarrollo de bases de conocimiento de entidades nombradas más que hacia su utilización en un sistema concreto.

Las bases de conocimiento estáticas no capturan toda la información necesaria para un dominio, especialmente aquellos que se encuentran en constante crecimiento. Por ejemplo, una lista con todas las compañías existentes en un país puede sufrir muchos cambios cada año. Por ello se hace necesario el uso de una base de conocimiento dinámica.

Una de las principales bases de conocimiento dinámicas actualmente en estudio es Wikipedia. Dado que es una enciclopedia colaborativa, la mayor parte de sus artículos son sobre entidades nombradas y tienen cierta estructura[53]. Un factor a tener en cuenta para determinar si una sucesión de *tokens* es una entidad puede ser la existencia o no de un artículo en Wikipedia con esa sucesión de *tokens* como título. Asimismo, un trabajo de minería en el texto del artículo[53] o la categoría del mismo[83] puede ayudar a determinar la clasificación de la entidad. Wikipedia posee páginas de desambiguación que contienen los posibles artículos a los que hace referencia una secuencia de *tokens* y la categoría a la que pertenecen. Por ejemplo, la página de desambiguación de *Habana* contiene referencias a las ciudades Habana en Cuba y Estados Unidos y a los municipios Centro Habana y Habana Vieja, además de a la película de Sydney Pollack con ese nombre. Un análisis del contexto puede determinar a cuál de estas entidades se hace referencia en el texto.

1.3.3. Sistemas basados en aprendizaje de máquinas

El aprendizaje de máquinas se emplea en el Reconocimiento de Entidades para inducir automáticamente sistemas basados en reglas. En este enfoque se emplean patrones y relaciones existentes en el texto para crear un modelo utilizando algoritmos de aprendizaje automático. Estos algoritmos se dividen en tres grandes grupos: supervisados, no supervisados y semi-supervisados.

Aprendizaje supervisado

El aprendizaje supervisado es actualmente el enfoque dominante para el Reconocimiento de Entidades[71]. El uso de esta técnica requiere una amplia colección de documentos con las menciones a entidades anotadas manualmente. Los algoritmos de aprendizaje supervisado estudian las par-

ticularidades de estas menciones. Este estudio permite a estos algoritmos inferir reglas que hacen posible la detección de nuevas entidades.

Entre las técnicas empleadas se incluyen algoritmos tradicionales como máquinas de soporte vectorial (SVM)[5], regresión logística[66], Adaboost[18] y árboles de decisión[88][86]. Estos han sido superados por algoritmos específicos para el aprendizaje de secuencias. Estos incluyen las cadenas de Markov (HMM)[8] y los modelos de máxima entropía (ME)[13]. Los algoritmos tradicionales requieren una cantidad considerable de características bien elegidas. Los modelos que aprenden de secuencias, en general, utilizan menos características. El campo aleatorio condicional (CRF)[65] es un modelo más reciente, utilizado también para segmentar y etiquetar secuencias de datos. Este algoritmo permite emplear un conjunto mucho más rico de atributos.

Estos clasificadores son entrenados empleando una serie de características no contextuales, léxicas, morfológicas y globales. Ejemplos de características no contextuales de un *token* son su posición en la secuencia analizada, su categoría gramatical, la presencia de signos de puntuación como puntos y comillas o el uso de mayúsculas y minúsculas. Entre las características léxicas se incluyen la frecuencia de aparición del *token* y si es o no un nombre propio. Las características globales pueden ser tomadas de bases de conocimiento o del resto del documento. En general, el éxito de los sistemas basados en aprendizaje supervisado depende de la elección de atributos tanto o más que de la elección del algoritmo de aprendizaje[86]. Los enfoques basados en aprendizaje supervisado pueden detectar entidades desconocidas para las bases de conocimiento[6].

Aprendizaje no supervisado

El aprendizaje no supervisado se distingue del supervisado por el hecho de que no requiere un conjunto de datos anotados. El objetivo del aprendizaje no supervisado es la construcción de representaciones de los datos que faciliten su clasificación. Este enfoque no es muy popular para el reconocimiento de entidades y los pocos sistemas que lo usan no son enteramente no supervisados[6].

La forma más común de abordar el aprendizaje no supervisado es mediante la conformación de *clusters*. Una variante consiste en agrupar conjuntos de *tokens* basados en la similitud entre sus contextos e identificar *clusters* representativos de cada tipo de entidad. Estos enfoques suelen mezclarse con el uso de bases de conocimiento como WordNet[71].

La mayor parte de los trabajos emplean aprendizaje no supervisado para la clasificación de entidades tras haberlas identificado utilizando otro méto-

do. Uno de estos trabajos asigna a cada *synset*[1] de *WordNet* un tipo de entidad[34]. A cada *synset* se asocia, además, un *contexto*. Este contexto no es más que el conjunto de palabras que aparecen con mayor frecuencia cerca de este en un corpus grande. Para determinar a qué clase pertenece una nueva palabra, se computa su contexto y se determina el *synset* cuyo contexto tiene mayor similitud con el suyo. La palabra se clasifica con el tipo de entidad asociado a ese *synset*.

Otro de estos trabajos identifica como entidades a toda secuencia de palabras con letra inicial mayúscula que aparecen en un documento[34]. Para clasificar la entidad, si X es la entidad identificada, se busca «tales como X» (*such as*) en un motor de búsqueda. El primer sustantivo que precede a la búsqueda se utiliza como categoría de la entidad.

Estos sistemas no tienen un conjunto predefinido de clases de entidades por lo que resulta difícil su evaluación. Sin embargo, recientemente se ha comprobado que el uso de características no supervisadas extraídas de corpus no anotados puede llevar a mejoras en otros sistemas de reconocimiento de entidades[98].

Aprendizaje semi-supervisado

El aprendizaje semi-supervisado utiliza datos de entrenamiento tanto etiquetados como no etiquetados: generalmente una pequeña cantidad de datos etiquetados junto a una gran cantidad de datos no etiquetados. El aprendizaje semi-supervisado se encuentra entre el aprendizaje no supervisado (sin datos de entrenamiento etiquetados) y el aprendizaje supervisado (con todos los datos de entrenamiento etiquetados). El costo asociado al proceso de anotación limita la disponibilidad de corpus etiquetados, mientras que la adquisición de datos sin etiquetar es relativamente poco costosa.

Las dos técnicas predominantes en el aprendizaje semi-supervisado son el *self-training*[19] y el *co-training*[11]. El *self-training* requiere un «conjunto semilla» anotado para iniciar el proceso de aprendizaje. El conjunto de datos sin anotar es etiquetado usando uno o más clasificadores. De los nuevos datos, los anotados con un mayor grado de confianza por los clasificadores son añadidos al conjunto semilla iterativamente. Por ejemplo, un sistema que busque «nombres de enfermedades» puede pedir al usuario un pequeño número de ejemplos. El sistema procede a buscar ejemplos de oraciones que contengan estos nombres e intenta identificar particularidades del contexto que sean comunes a estos ejemplos. Una vez completada esta etapa, el siste-

[1] Conjunto de palabras con similar significado

ma buscará instancias que aparecen en contextos similares. Este proceso es repetido con las nuevas instancias para encontrar nuevos contextos. Algunos sistemas utilizan herramientas de reconocimiento de entidades existentes para elaborar el conjunto inicial de entidades[27].

El *co-training* es una extensión del *self-training*. En esta técnica se entrenan dos o más sistemas. Cada sistema utiliza un conjunto diferente de características. Esta técnica solo es efectiva si los conjuntos de características son independientes y cada conjunto es suficiente para predecir correctamente la clase del *token*[58]. Las predicciones más acertadas son usadas para construir el conjunto de entrenamiento iterativamente.

Capítulo 2

Propuesta de Metodología para el Reconocimiento de Entidades Nombradas

Los sistemas basados en reglas no se comportan bien en ambientes con diversidad de ortografías y formatos. Esto implica que no son adecuados para el Reconocimiento de Entidades en *Twitter*. Los sistemas basados enteramente en bases de conocimiento tampoco son adecuados. Esto se debe a que incluso una enciclopedia tan dinámica como Wikipedia, con más de 300 artículos nuevos publicados cada día en su edición en español[100], no es capaz de recoger todas las entidades que se mencionan en una red social de *microblogging* como *Twitter*. Por esto, el enfoque que se utiliza en este trabajo es el basado en aprendizaje de máquinas.

No existen trabajos que avalen el uso de aprendizaje no supervisado como único mecanismo para el Reconocimiento de Entidades de diversas clases en un dominio amplio. El aprendizaje supervisado, por su parte, requiere un gran volumen de ejemplos de entrenamiento anotados para ser efectivo[73]. Estos procesos de anotación requieren tiempo y la presencia de expertos. A esto se suma que un sistema desarrollado empleando aprendizaje supervisado no es fácilmente portable a un nuevo dominio.

En este trabajo se utiliza el aprendizaje semi-supervisado. Este enfoque posee varias de las características positivas que colocan al aprendizaje supervisado como el enfoque más utilizado en la actualidad para el reconocimiento de entidades, pero requiere un menor número de ejemplos de entrenamiento. El algoritmo empleado es *self-training*, pues demostrar que dos conjuntos de características son independientes y suficientes, pre-requisito del *co-training*,

es una tarea compleja. Como clasificadores se usan algoritmos de aprendizaje tradicionales y algoritmos de aprendizaje de secuencias.

Con el fin de proveer conocimiento externo a los clasificadores se utilizan *wikis*. Esto, unido al uso de características no supervisadas extraídas de corpus no anotados, contribuirá a la portabilidad de la propuesta.

Las estrategias para el Reconocimiento de Entidades basados en aprendizaje de máquinas constan de cuatro etapas, como se muestra en la figura 2.1.

Figura 2.1: Etapas del Reconocimiento de Entidades con aprendizaje de máquinas

- **Preprocesamiento:**

  Es el proceso de normalización del texto. Elimina irregularidades de diverso tipo para mejorar los resultados de los algoritmos de aprendizaje que serán aplicados posteriormente.

- **Selección de características:**

  Es el proceso de identificación de los atributos de un *token* que son relevantes para el reconocimiento de entidades.

- **Selección de clasificadores:**

  Es el proceso de selección de los algoritmos de aprendizaje de máquinas más adecuados para el problema en cuestión.

- **Clasificación:**

  Es el proceso de aprender de la lista de características para detectar y clasificar nuevas entidades.

2.1. Preprocesamiento

Usualmente, el resultado de los algoritmos de aprendizaje de máquinas en la extracción de información se mejora mediante un preprocesamiento del texto[35]. Sin embargo, existen pocas menciones en la literatura a esta fase inicial de la extracción de información en el Reconocimiento de Entidades. A continuación se analizan diversas etapas del preprocesamiento de textos que se emplean en otras áreas de la minería de textos y su pertinencia para esta tarea en particular.

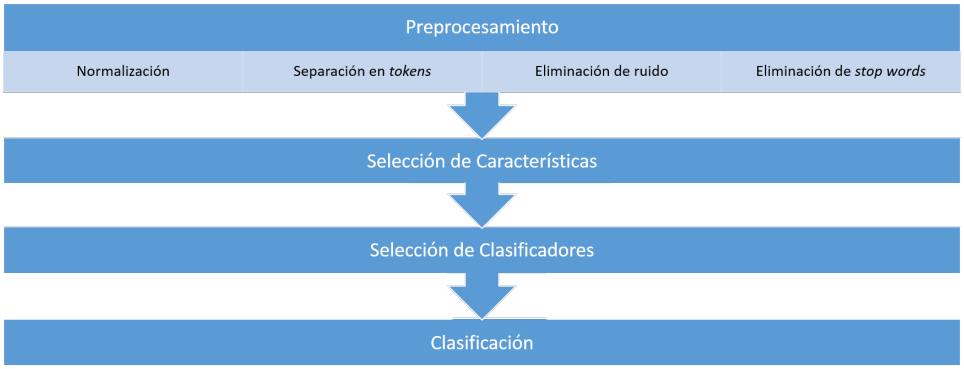

Figura 2.2: Etapas del preprocesamiento de textos

2.1.1. Separación en tokens

Esta fase consiste en la separación de un texto en palabras, frases, símbolos u otra unidad sintáctica llamada *token*. La lista de *tokens* es la entrada para las etapas posteriores del análisis. En este caso, cada *tweet* debe ser separado en una secuencia de palabras. Los modelos clásicos de tokenización como *Penn Treebank* (*PTB*)[94] no tienen buenos resultados al intentar separar *tweets* en *tokens* debido a la presencia de *emoticonos* y URLs[12]. La herramienta que más se utiliza en la literatura con este fin es *twokenize*[76][75].

2.1.2. Normalización

Esta fase es un paso común en muchas tareas del procesamiento de lenguaje natural. La normalización del texto consiste generalmente en la conversión del texto a minúsculas y el *stemming*. La conversión a minúsculas elimina uno de los atributos que caracterizan a las entidades nombradas: las

mayúsculas[84]. El *stemming* consiste en la eliminación de prefijos y sufijos comunes. Esto, generalmente, tiene un efecto no deseado en los nombres propios por lo que también puede resultar contraproducente para el reconocimiento de entidades[63]. Atendiendo a esto, esta etapa del preprocesamiento no se incluye en la metodología.

2.1.3. Eliminación de ruido

Esta fase consiste en la eliminación o sustitución de rasgos presentes en un *tweet* que no aparecen en textos formales. Entre estos rasgos se encuentran *emoticonos*, abreviaturas, jerga y errores ortográficos. Ambientes como las redes sociales de *microblogging* son un reto para las herramientas existentes de eliminación de ruido.

Uno de los pasos para la eliminación de ruido en *tweets* es la supresión de marcas propias de *Twitter* como las menciones a usuarios y los *hashtags*. Esto no es apropiado para un sistema de reconocimiento de entidades, pues muchos *hashtags* y menciones a usuarios son, también, menciones a entidades nombradas, como se muestra en la tabla 2.1. Algunos trabajos anteriores obvian las menciones a usuarios[85] y otros las catalogan siempre como una entidad de tipo persona[84].

Clasificación	**Hashtag**	**Mención a usuario**
Persona	#XiJinping #EnriquePeñaNieto	@BarackObama @camila_vallejo
Organización	#ColumbiaUniversity #UNASUR	@CELAC @JuventudRebelde
Lugar	#Havana #África	
Misceláneas	#Feriadellibro #Brooklynprotest	@ChampionsLeague @WBCBaseball
No entidad	#periodistas #Educación	

Tabla 2.1: Ejemplos de hashtags y menciones a usuarios con sus clasificaciones

Otro paso es el uso de correctores ortográficos para sustituir la jerga y las abreviaturas y enmendar los errores ortográficos. Estas herramientas

tienden a corregir palabras correctamente escritas, pero que no aparecen en diccionarios, como los nombres propios. Esto es muy poco deseable en un sistema de reconocimiento de entidades donde es imprescindible reconocer nombres propios. Además, la mejoría reportada al usar herramientas de eliminación de ruido en otros trabajos es escasa[33][29]. Por estas razones, en esta propuesta se prescinde de esta etapa del preprocesamiento.

2.1.4. Eliminación de stop words

Las *stop words* son palabras con poco aporte semántico y que suelen eliminarse durante el preprocesamiento de los *tweets*. Por una parte, la eliminación de *stop words* permite al sistema hacer énfasis en los *tokens* de mayor interés y reduce considerablemente las dimensiones del problema. Por otra parte, al eliminar estos *tokens*, se rompe la secuencia de palabras del *tweet*. Esto puede resultar negativo para el Reconocimiento de Entidades. En el capítulo 3 se evalúa el impacto de esta fase en los resultados de la clasificación.

2.2. Selección de características

La elección del mejor conjunto de atributos para representar una entidad afecta varios aspectos del Reconocimiento de Entidades como la precisión, el tiempo del aprendizaje y el tamaño del conjunto de entrenamiento.

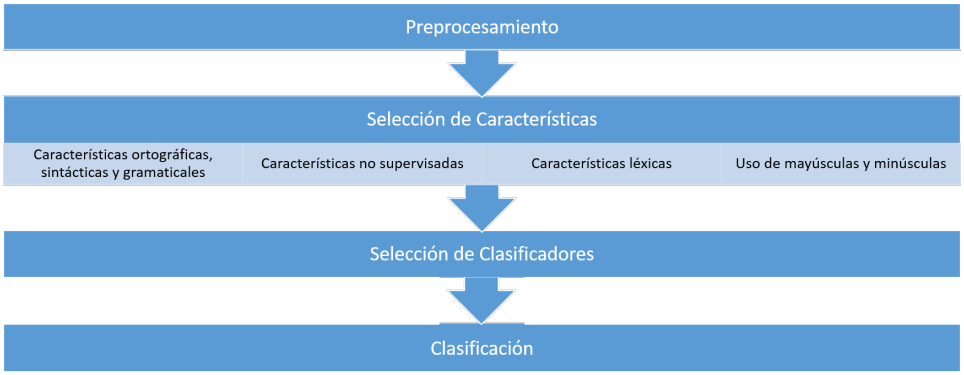

Figura 2.3: Grupos de características de un *token*

En muchas aplicaciones del Reconocimiento de Entidades es común encontrar cientos o miles de características. Es por esto que es común buscar

estrategias para eliminar características redundantes e irrelevantes, e identificar el subconjunto óptimo de características a utilizar. Se han propuesto diversos enfoques para la selección de un conjunto de características que optimice la precisión del sistema y el tiempo de aprendizaje como el uso de algoritmos genéticos[56] y medidores como la ganancia de información[95].

En otros trabajos se utilizan todas las características disponibles[18][91][103]. Dado que el enfoque que se emplea en este trabajo es el aprendizaje semisupervisado con un conjunto de entrenamiento pequeño, se utilizan todas las características. El algoritmo de aprendizaje es el encargado de ignorar los atributos poco relevantes para el problema.

2.2.1. Características ortográficas, sintácticas y gramaticales

Las primeras características que se propone utilizar son atributos ortográficos, sintácticos y gramaticales del *token* y los *tokens* presentes en una vecindad de este. Entre estas se encuentran la secuencia de caracteres del *token*, el uso de mayúsculas y minúsculas en este, sus prefijos y sufijos y su categoría gramatical. La lista completa de características ortográficas y gramaticales empleadas para el aprendizaje puede encontrarse en la tabla 2.2.

El mapeo empleado para representar el uso de mayúsculas y minúsculas en un *token* consiste en la sustitución de todas las minúsculas por a, las mayúsculas por A, los dígitos por 0 y los caracteres restantes por $-$. De esta forma: *IBM* se convierte en AAA, *Cuba* en $Aaaa$ y *Mundial-2014* en $Aaaaaaa\text{-}0000$. En un segundo paso se reemplazan las secuencias de más de un caracter de un mismo tipo por dos repeticiones del caracter como se muestra en la figura 2.4.

$$IBM \Longrightarrow AAA \Longrightarrow AA$$

$$Cuba \Longrightarrow Aaaa \Longrightarrow Aaa$$

$$Mundial\text{-}2014 \Longrightarrow Aaaaaaa\text{-}0000 \Longrightarrow Aaa\text{-}00$$

Figura 2.4: Mapeo empleado para representar el uso de mayúsculas y minúsculas en un *token*

La categoría gramatical de un *token* es un atributo importante para un gran número de tareas del procesamiento de lenguaje natural incluyendo el Reconocimiento de Entidades. El etiquetado gramatical (conocido también

Característica	**Descripción**
t_0	token actual
t_{-1}, t_{+1}	token precedente y token siguiente
$t_0.lower$	token en minúsculas
$t_0[:1]$, $t_0[:2]$, $t_0[:3]$	prefijos de tamaño uno, dos y tres
$t_0[-1:]$, $t_0[-2:]$, $t_0[-3:]$	sufijos de tamaño uno, dos y tres
$t_0.isupper$	indica si el token está compuesto solo por mayúsculas
$t_0.istitle$	indica si el token comienza con una mayúscula
$t_0.isdigit$	indica si el token está compuesto únicamente por dígitos
$t_0.shape$	mapeo de un token a una expresión que representa el uso de mayúsculas y minúsculas en el token
$t_{-1}.shape, t_{+1}.shape$	
$t_0.postag$	categoría gramatical del token
$t_{-1}.postag, t_{+1}.postag$	
$t_{-1}\&t_0$	concatenación del token precedente y el actual
$t_0\&t_1$	concatenación del token actual y el siguiente
$t_0.first$	indica si el token es el primero del tweet
$t_0.last$	indica si el token es el último del tweet

Tabla 2.2: Características ortográficas, sintácticas y gramaticales empleadas

por su nombre en inglés, *part-of-speech tagging* o *POS tagging*) tiene una *línea de base* muy fuerte: un algoritmo que asigna a cada palabra del vocabulario su etiqueta más frecuente y a cada palabra que no aparece en el vocabulario la etiqueta más común obtiene una precisión del 90 % en corpus noticiosos[20]. Sin embargo, el empleo de este mismo algoritmo en *tweets* lleva a la obtención de una precisión del 76 %[84]. Un factor que influye en esto es el gran número de palabras que no aparecen en el vocabulario que hay en los *tweets*. El Stanford POS Tagger[42], el tagger de mejores resultados reportados en la literatura, mejora estos resultados en tweets hasta un 80 %, resultado que dista mucho del 97 % logrado por el tagger en el corpus para el que fue desarrollado[84][96]. Debido a la complejidad de la asignación de una categoría gramatical a un token, en el capítulo 3 se evalúa el impacto de prescindir de esta fase en los resultados de la clasificación.

2.2.2. Características no supervisadas

Además de los atributos ortográficos, sintácticos y gramaticales de un *token* se propone emplear características obtenidas utilizando aprendizaje no supervisado. Algunos trabajos emplean datos anotados y datos no anotados en su proceso de entrenamiento[4][93]. Sin embargo, una manera más sencilla de mejorar los resultados de un sistema de reconocimiento de entidades es utilizar atributos obtenidos a partir de datos no anotados como características en el aprendizaje supervisado[80][98]. Para esto, se emplean tres algoritmos de *clustering*.

Clusters de Brown

Existen conjuntos de palabras similares en su significado y función sintáctica. Estas palabras suelen aparecer en contextos similares. El objetivo de los *clusters* de Brown es agrupar en una misma clase palabras que aparecen en contextos similares[16]. Esto permite hacer predicciones sobre el significado de palabras no vistas durante el entrenamiento encontrando similitudes con palabras que ya se han visto. Los *clusters* de Brown suelen asignar entidades de un mismo tipo a una misma clase[95] por lo que añadir el *cluster* al que pertenece una palabra a la lista de características de un *token* puede ayudar en su clasificación.

La salida del algoritmo es un *dendograma*. Un camino partiendo de la raíz representa una palabra codificada en una secuencia binaria. Fijando una longitud de prefijo se divide el vocabulario en clases. El número de clases es igual al número de prefijos diferentes de la longitud fijada.

Clusters de Clark

Los *clusters* de Clark agrupan palabras que tienen un contexto similar comenzando por las más frecuentes[22]. El algoritmo es similar al empleado para obtener *clusters* de Brown, pero tiene en cuenta, además del contexto de la palabra, sus características morfológicas. También tiene en cuenta la frecuencia de aparición de los términos. Esto puede ser relevante para el Reconocimiento de Entidades, atendiendo a que las palabras que aparecen con poca frecuencia suelen ser nombres propios y menciones a entidades con mayor probabilidad que las que aparecen con mayor frecuencia.

Word2vec

El algoritmo *word2vec*[62][69] fue creado por un grupo de investigación de *Google*[1]. Es un conjunto de modelos utilizado para generar *clusters* de palabras. Estos modelos son redes neuronales de dos capas entrenadas para reconstruir el contexto de una palabra. *Word2vec* recibe como entrada un gran corpus de texto y produce como salida un espacio vectorial donde a cada palabra del corpus se le asigna un único vector. Los vectores de palabras están distribuidos de modo tal que palabras que comparten contextos similares son cercanas en el espacio. La distancia entre vectores es medida utilizando el coseno del ángulo comprendido entre estos.

2.2.3. Características globales

Las características globales son atributos que se asignan a un *token* después de consultar una base de conocimiento externa. Se ha demostrado que el uso de bases de conocimiento y diccionarios de entidades es importante para mejorar el rendimiento de los sistemas de reconocimiento de entidades[53]. Sin embargo, construir y mantener bases de conocimiento de calidad es un proceso costoso. En este trabajo se propone el uso de Wikipedia con este fin.

Wikipedia no es una base de conocimiento, pues no está diseñada para ser procesada automáticamente. Sin embargo, extraer información de Wikipedia es mucho más sencillo que extraerla de texto plano debido a que la mayor parte de sus artículos siguen un patrón o estructura.

El primer paso de la propuesta es hallar todas las colocaciones terminológicas presentes en el corpus de *tweets*. Las colocaciones son expresiones de más de una palabra que aparecen juntas con frecuencia. El primer atributo

[1] `www.google.com`

global de un *token* se obtiene verificando la existencia o no de una página en Wikipedia cuyo título coincida con el *token* o una colocación donde aparezca este. El atributo toma valor *O* si no se encuentra la página, *B* si el *token* se encuentra al inicio del título de la página encontrada e *I* si se encuentra en el título pero no como primera palabra.

El segundo atributo se explora solo si se encuentra la página. Se obtiene consultando las categorías a las que pertenece la página de Wikipedia. De esta lista de categorías se eliminan aquellas que incluyen las palabras *artículo*, Wikipedia o *página*, pues estas categorías tienen que ver con la estructura administrativa de Wikipedia y no tienen aporte semántico. Se añade como atributo, de la lista de categorías, aquella que se repite con mayor frecuencia en el corpus.

El tercer atributo, al igual que el segundo, se explora solo si se encuentra la página. A pesar de que no existe un formato estricto para los artículos de la enciclopedia colaborativa, es convención comenzar el artículo con una breve oración que define la entidad descrita en el artículo. Se utiliza como atributo el primer sustantivo que aparece tras el verbo *ser* en cualquiera de sus conjugaciones.

Por ejemplo, en el *tweet* que se muestra en la figura 2.5, al analizar el *token* «Benny» se identifica que este aparece en la colocación «Benny Moré». Esto nos llevará a la página de Wikipedia: `es.wikipedia.org/wiki/Benny_Moré`. Dado que existe la página en Wikipedia y «Benny» es la primera palabra del título de la página, el primer atributo global del *token* «Benny» toma valor *B*.

Como se ve en la figura 2.6, el artículo sobre Benny Moré pertenece a nueve categorías diferentes. En este caso, «Hombres» es la que aparece con mayor frecuencia por lo que el segundo atributo global del *token* «Benny» es «Hombres».

Como se ve en la figura 2.7, «fue» es la primera forma del verbo ser que aparece en el artículo y «cantante» es el primer sustantivo que aparece tras esta. El tercer atributo global del *token* «Benny» es «cantante».

2.2.4. Uso de mayúsculas

En algunos tweets el uso de mayúsculas es el adecuado y por tanto resulta informativo, en otros casos resulta engañoso. Algunos tweets están escritos enteramente en mayúsculas (0.6 %), otros están enteramente en minúsculas [84] (8 %).

Se añade a cada *tweet* un atributo binario que indica si el uso de mayúsculas resulta informativo. Los criterios empleados para determinar si el uso

Figura 2.5: *Tweet* sobre Benny Moré

Figura 2.6: Categorías del artículo sobre Benny Moré en Wikipedia

de mayúsculas es adecuado son:

- El *tweet* no está enteramente en mayúsculas ni enteramente en minúsculas.
- El *tweet* comienza con mayúsculas.
- A un punto final (.) le sigue una mayúscula.
- Menos del 50 % de los caracteres son mayúsculas.
- Palabras que en un corpus estructurado aparecen con mayúsculas, están escritas de esa manera en el *tweet*.

Figura 2.7: Artículo sobre Benny Moré en Wikipedia

- Palabras que en un corpus estructurado aparecen con minúsculas, están escritas de esa manera en el *tweet*.

2.3. Selección de clasificadores

Para el proceso de clasificación se utilizan algoritmos de dos tipos: algoritmos de aprendizaje supervisado y algoritmos de aprendizaje de secuencia o de datos estructurados. Dado un conjunto de *token* representados mediante las características mencionadas en la sección 2.2, un algoritmo de aprendizaje supervisado debe producir una función de inferencia. Esta función debe ser capaz, idealmente, de determinar la clase correcta para *tokens* no vistos con anterioridad.

2.3.1. Clasificadores tradicionales

- **Naive Bayes**

  Naive Bayes es un algoritmo probabilístico de aprendizaje supervisado basado en la aplicación del teorema de Bayes. Este clasificador asume

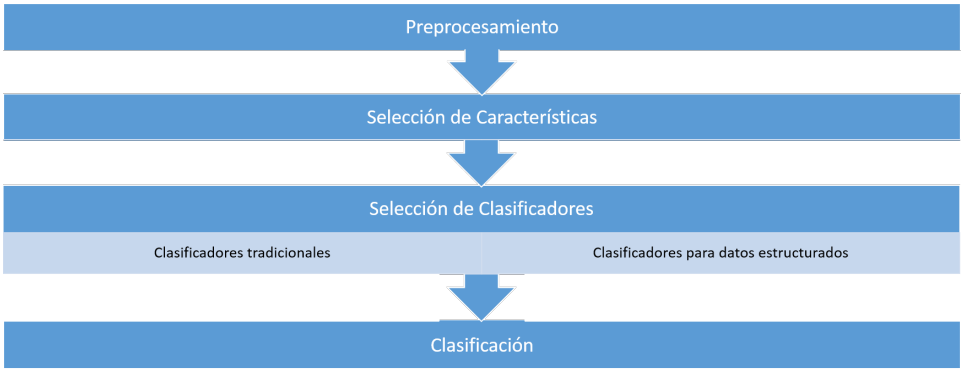

Figura 2.8: Tipos de clasificadores para el Reconocimiento de Entidades

que el valor de una característica particular es independiente del valor de cualquier otra característica. Una ventaja de Naive Bayes es que no requiere una gran cantidad de datos de entrenamiento para estimar los parámetros necesarios para la clasificación[101].

- **Árboles de Decisión**

  Los Árboles de Decisión tienen como objetivo construir un árbol que explique cada ejemplo de entrenamiento de la manera más compacta posible. Los nodos internos son etiquetados como atributos, las ramas salientes de cada nodo representan pruebas para los valores del atributo, y las hojas del árbol identifican a las categorías. Una ventaja de los Árboles de Decisión es que son resistentes al ruido. Sin embargo, las particiones del árbol pueden ser muy específicas para el conjunto de entrenamiento[15].

- **Support Vector Machine (SVM)**

  SVM busca una superficie que separe el conjunto de entrenamiento en el espacio de modo tal que los ejemplos pertenecientes a categorías distintas estén separados por el margen más amplio posible. Se basa en un clasificador lineal muy sencillo, precedido de una transformación del espacio (a través de un núcleo) para darle potencia expresiva. Cuando el número de características es mucho mayor que el número de ejemplos de entrenamiento, sus resultados son pobres[17].

- **Stochastic Gradient Descent (SGD)**

SGD es una aproximación estocástica al descenso por gradientes. El descenso por gradientes minimiza una función objetivo iterativamente tomando pasos proporcionales al gradiente negativo de la función en el punto actual. El descenso por gradientes realiza cálculos redundantes. SGD remedia esto procesando los datos de entrenamiento uno a la vez, por lo que es usualmente más rápido que el descenso por gradientes. Ha sido usado con éxito en el aprendizaje a gran escala[102].

- **Perceptrón**

  El perceptrón es la más sencilla de las redes neuronales, consta de una única capa. El perceptrón es entrenado para clasificar los ejemplos de entrenamiento correctamente ajustando los pesos asociados a cada característica. Es adecuado para el aprendizaje a gran escala[40].

- **Passive Aggressive Classifier (PAC)**

  PAC es un modelo de clasificación similar al perceptrón. Además de garantizar que todos los ejemplos de entrenamiento se clasifiquen correctamente, PAC garantiza la existencia de un margen entre los ejemplos pertenecientes a categorías distintas. Es adecuado para el aprendizaje a gran escala[25].

- **Ensemble Methods**

  Los métodos de agregación de modelos de aprendizaje automático combinan varias hipótesis hechas sobre un mismo conjunto de datos con el fin de obtener un modelo predictivo con un mejor rendimiento. Existen dos conjuntos de métodos de agregación. Los métodos de promedio combinan varios estimadores promediando sus predicciones. Los métodos de *boosting* se construyen secuencialmente. Cada clasificador intenta reducir la parcialidad de los clasificadores que le preceden en la clasificación. Los principales *Ensemble Methods* son:

  - **AdaBoost**

    AdaBoost es un método de agregación usando *boosting*. Emplea varios modelos de aprendizaje sencillos, que son solo un poco mejores que la clasificación aleatoria. Un ejemplo de esto son los árboles de decisión pequeños. A cada uno de estos clasificadores simples se le proporciona una versión modificada de los datos. La modificación que se realiza a los datos consiste en aplicar pesos distintos a cada uno de los ejemplos de entrenamiento. En cada paso del algoritmo el peso asociado a los ejemplos que han sido

correctamente clasificados se incrementa y se decrementa el peso asociado a los erróneamente clasificados. Las predicciones de los clasificadores se combinan para producir la clasificación final[39].

- **Random Forest**
  Los *Random Forest* son un método de agregración usando promedio. El modelo construye árboles de decisión con distintos subconjuntos de los ejemplos de entrenamiento. Esto hace que la parcialidad del modelo incremente. Promediar los resultados obtenidos por cada árbol de decisión hace que disminuya la varianza y compensa el alza en la parcialidad del modelo. Muchas veces se obtiene un mejor resultado que con un único árbol de decisión[14].

2.3.2. Clasificadores para datos estructurados

Los clasificadores para datos estructurados o de aprendizaje de secuencias buscan encontrar la secuencia de clases de entidades más probable dada una secuencia de *tokens*. Los tres clasificadores de este tipo que se emplean en este trabajo son:

- **Cadenas Ocultas de Markov (HMM)**
  Las Cadenas Ocultas de Markox fueron el primer algoritmo de aprendizaje de secuencias utilizado para el Reconocimiento de Entidades en 1999[9]. Es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos u ocultos de dicha cadena a partir de los parámetros observables. Se modela mediante un autómata finito[31].

- **Modelo de Máxima Entropía**
  Un Modelo de Máxima Entropía es aquel que satisface el Principio de Máxima Entropía[51] que establece que una correcta distribución es aquella que maximiza la entropía o incertidumbre respetando las restricciones (los hechos conocidos). Intuitivamente, es un modelo que, siendo consistente con los datos, es tan uniforme como es posible. Este enfoque se utiliza con frecuencia cuando no debe asumirse independencia entre *tokens*[7].

- **Campos Aleatorios Condicionales (CRF)**
  Un Campo Aleatorio Condicional es un modelo estocástico utilizado habitualmente para etiquetar y segmentar secuencias de datos o extraer información de documentos. En algunos contextos también se

lo denomina campo aleatorio de Markov. Los Campos Aleatorios disminuyen las suposiciones de independencia presentes en las Cadenas Ocultas de Markov. También eliminan una de las principales limitaciones de los Modelos de Máxima Entropía: la tendencia de favorecer los estados con pocos sucesores[59].

Capítulo 3

Experimentación

En este capítulo se presenta un conjunto de experimentos realizados para medir la eficacia de las características y algoritmos de clasificación propuestos. Seguidamente, se muestran los resultados obtenidos por el algoritmo de *self-training*. Se realiza una comparación con otros sistemas de Reconocimiento de Entidades Nombradas y se muestran los resultados obtenidos al migrar el sistema a otro dominio. Todos los experimentos realizados fueron repetidos en 30 ocasiones. En los experimentos se utiliza validación cruzada (*k-fold*)[47] con cuatro particiones.

3.1. Corpus

El corpus con el que se realizaron las evaluaciones iniciales es el *xLiMe Twitter Corpus*[82]. Este corpus contiene *tweets* en italiano, alemán y español. Los *tweets* están tokenizados y tienen etiquetas que indican la categoría gramatical y el tipo de entidad por cada *token*. Estas anotaciones fueron realizadas manualmente.

El corpus tiene un total de 20000 mensajes. De estos, 7713 son en español. Los experimentos se realizan utilizando validación cruzada con cuatro particiones por lo que se utilizan 5785 tweets para el entrenamiento. de Los *tweets* en español tienen un total de 140852 *tokens*. En la tabla 3.1 se muestra el número de entidades de cada tipo presente en el corpus.

Las entidades del corpus están separadas en cuatro categorías: persona, lugar, organización y misceláneas. Siguiendo el estándar *BIO* un *token* tiene asociado una de las nueve clases listadas en la tabla 3.2.

Tipo de Entidad	**Alemán**	**Italiano**	**Español**
Lugar	742	2087	1441
Misceláneas	995	5802	775
Organización	350	1150	836
Persona	757	3701	2321
Total	2844	12740	5373

Tabla 3.1: Menciones a entidades nombradas en el *xLiMe Twitter Corpus*

Clase	**Descripción**
B-PER	Primer token de una entidad de tipo persona
I-PER	Token que pertenece a una entidad de tipo persona, pero no es su primer token
B-LOC	Primer token de una entidad de tipo lugar
I-LOC	Token que pertenece a una entidad de tipo lugar, pero no es su primer token
B-ORG	Primer token de una entidad de tipo organización
I-ORG	Token que pertenece a una entidad de tipo organización, pero no es su primer token
B-MISC	Primer token de una entidad que no es de ninguno de los tipos anteriores
I-MISC	Token que pertenece a una entidad de tipo miscelánea, pero no es su primer token
O	Token que no pertenece a una entidad

Tabla 3.2: Posibles clasificaciones para un token en el *xLiMe Twitter Corpus*

3.2. Detalles de implementación

3.2.1. Selección de características

En total, para cada *token*, se definen 33 características. La lista de todos los atributos se muestra en el Anexo 1. Al aplicar el algoritmo de *Feature Hashing*[99] se obtiene una matriz de 140852 filas (el número de *tokens*) y casi tres millones columnas. En la sección 3.2.3 se valora el efecto de disminuir las dimensiones de la matriz.

Categoría gramatical

El corpus empleado en este trabajo tiene asociado a cada *token* una categoría gramatical. Esta asociación es realizada manualmente. Es posible, que en caso de no contarse con etiquetas manuales y debido a la complejidad del *pos-tagging*, sea conveniente eliminar este atributo del conjunto de características. En el Anexo 2 se listan todas las categorías gramaticales presentes en el corpus. En la sección 3.2.3 se evalúa el impacto de este atributo.

Uso de mayúsculas

Para determinar si el uso de mayúsculas y minúsculas en un *token* del *tweet* es el apropiado se utiliza una porción del *Wikicorpus*[81] en español de más de 120 millones de palabras. Para esto, se compara la forma en que está escrito el *token* en el *tweet* con las formas en que aparece en el *Wikicorpus*.

Características no supervisadas

Para computar los *clusters* de palabras, se une al *xLiMe Twitter Corpus* un corpus no anotado de un millón de *tweets* en español obtenidos mediante el *API* de *Twitter*[1]. Empleando los algoritmos de Clark[22] y *Word2vec*[62][69], se dividen los *tokens* en 100 *clusters*. En el caso del algoritmo de Brown[16], se hace uso de su salida en forma de dendograma para dividir los *tokens* en 50, 80 y 100 *clusters*. En la figura 3.1 se muestran cuatro *clusters* obtenidos donde predominan entidades de tipo persona, entidades de tipo lugar, entidades de tipo organización y palabras que no son entidades (principalmente formas verbales).

[1] `https://dev.twitter.com/rest/public`

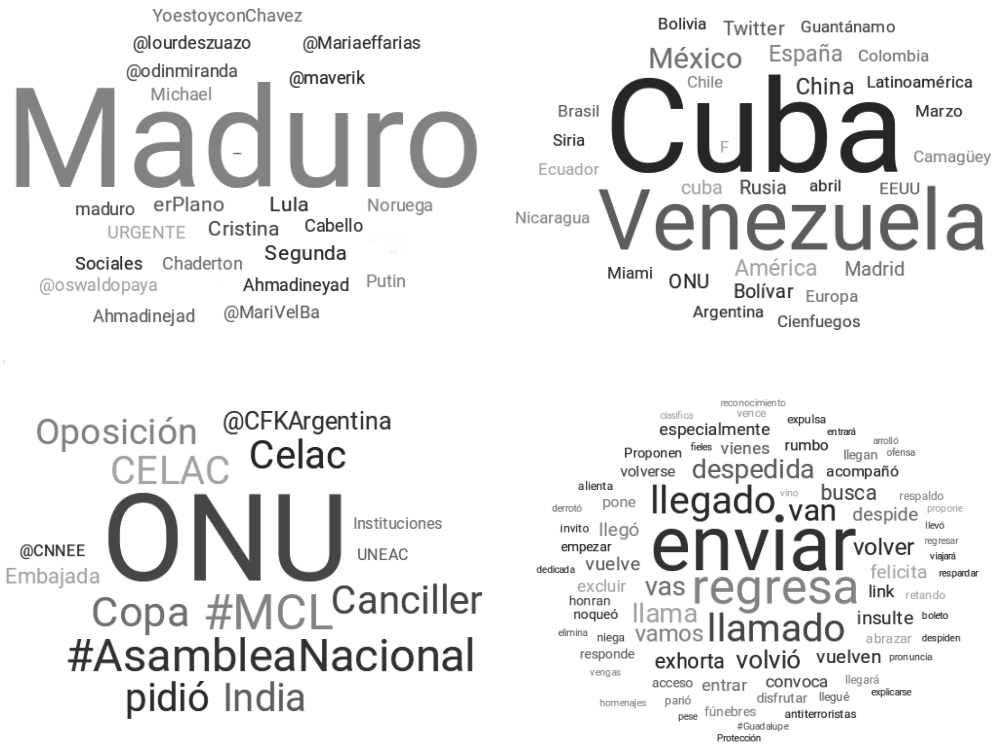

Figura 3.1: Representación mediante nubes de etiquetas de cuatro *clusters* de palabras

3.2.2. Selección de clasificadores

Clasificadores tradicionales

Para la comparación se utilizan las implementaciones de *scikit-learn*[77] de Naive Bayes, Árboles de Decisión, SVM, SGD, Perceptron, PAC, AdaBoost y Random Forest. Los porcientos de precisión y exhaustividad son superiores al 95 % para todos los clasificadores. Estos resultados se deben a que solo 2844 *tokens* de los 140852 son entidades para un 0.02 %. Cualquier modelo que asigne la mayor parte de los *tokens* a la categoría *O*, obtendrá una medida F1 elevada. Es por esto que a partir de este momento se muestran la precisión, exhaustividad y medida F1 teniendo en cuenta solo las clases que representan entidades.

En la tabla 3.3 se muestran los resultados obtenidos. Se usa como línea

Clasificador	**Precisión**	**Exhaustividad**	**Medida F1**
Línea de base	0.006	0.006	0.006
Naive Bayes	0.552	0.266	0.324
Árboles de Decisión	0.320	0.401	0.352
SVM	0.283	**0.553**	0.370
SGD	**0.589**	0.412	0.457
Perceptron	0.481	0.435	0.436
PAC	0.533	0.426	**0.458**
AdaBoost	0.479	0.308	0.348
Random Forest	0.564	0.303	0.383

Tabla 3.3: Precisión, exhaustividad y medida F1 por clasificador

de base para comparar los clasificadores un algoritmo aleatorio que asigna a cada *token* una clase aleatoriamente, respetando la distribución de los datos en el conjunto de entrenamiento. Puede verse que los algoritmos con mejores resultados en cuanto a medida F1 son PAC, SGD y Perceptron en ese orden. SVM es el que mejor exhaustividad tiene, pero es también el de menor precisión.

En la tabla 3.4 se muestran los resultados por clase obtenidos por PAC, el clasificador con mejores resultados de este primer grupo. En la figura 3.2 se muestra la matriz de confusión obtenida. Se puede ver que las entidades de tipo persona y lugar son las más sencillas de clasificar, seguidas de las de tipo organización y por último las misceláneas. Esto es congruente con lo visto en otros sistemas de reconocimiento de entidades[29]. Las entidades de tipo *I-X* son clasificadas correctamente con menor precisión y exhaustividad que las de tipo *B-X*.

Clasificadores para datos estructurados

Para la clasificación usando este tipo de clasificadores se utilizaron las distintas variantes de CRF disponibles en *sklearn-crfsuite*[26]. Este es una capa de abstracción para *Python* de *CRFsuite*[74]. Las variantes de CRF utilizan distintos algoritmos de entrenamiento: Limited-memory BFGS (L-BFGS), Stochastic Gradient Descent (L2-SGD)[90], Averaged Perceptron[24], Passive Aggressive[25] y Adaptive Regularization Of Weight Vector (AROW)[68]. La implementación de Modelos de Máxima Entropía empleada es la disponible en *NLTK*[10]. La implementación de HMM es la disponible en *hmmlearn*[49]. En la tabla 3.5 se muestran los resultados obtenidos por los

Tipo de Entidad	**Precisión**	**Exhaustividad**	**Medida F1**
B-LOC	0.624	**0.590**	**0.599**
B-MISC	0.348	0.160	0.210
B-ORG	0.495	0.326	0.383
B-PER	**0.636**	0.569	0.595
I-LOC	0.354	0.178	0.227
I-MISC	0.314	0.114	0.160
I-ORG	0.353	0.153	0.208
I-PER	0.608	0.534	0.555
O	0.986	0.995	0.990

Tabla 3.4: Precisión, exhaustividad y medida F1 por tipo de entidad obtenidos utilizando PAC

algoritmos para clasificación de datos estructurados.

Clasificador	Precisión	Exhaustividad	Medida F1
CRF (L-BFGS)	0.626	0.475	**0.517**
CRF (L2-SGD)	**0.635**	0.459	0.509
CRF (AP)	0.548	**0.465**	0.490
CRF (PA)	0.557	0.547	0.488
CRF (AROW)	0.481	0.412	0.434
Maximum Entropy	0.520	0.381	0.427
HMM	0.054	0.136	0.077

Tabla 3.5: Precisión, exhaustividad y medida F1 por clasificador para datos estructurados

Puede observarse una mejoría respecto a los algoritmos tradicionales. Es llamativa la baja medida F1 reportada al utilizar HMM. Esto puede deberse al gran número de características utilizado. El clasificador con mejores resultados es CRF (L-BFGS). En la tabla 3.6 se muestran las transiciones de una clase a otra que ocurren con mayor probabilidad según dicho clasificador. Puede verse que las transiciones más probables son aquellas que van de *B-X* a *I-X* y las que van de *I-X* a *I-X* lo cual indica que un gran número de entidades están compuestas por más de dos *tokens*.

En la tabla 3.7 se muestran las transiciones que ocurren con menor probabilidad. Las transiciones menos probables son las que van de *O* a *I-X* dado que *I-X* siempre debe estar precedida por *B-X*. También es poco probable

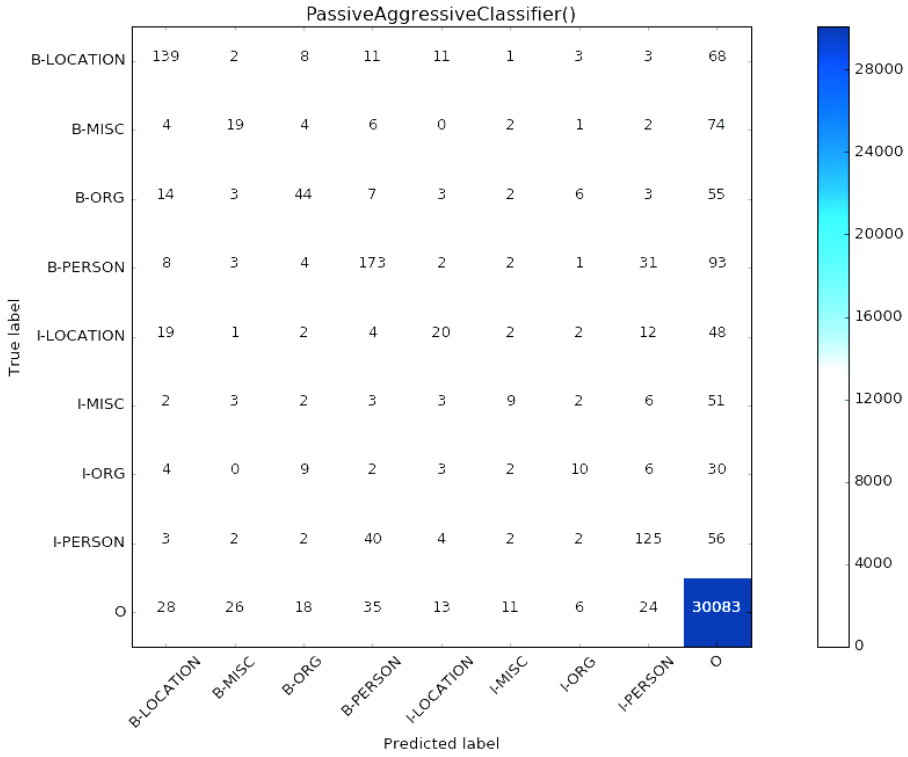

Figura 3.2: Matriz de confusión obtenida utilizando PAC

una transición de *B-LOC* a *O*, lo cual es señal de que las entidades de tipo lugar suelen contener más de un *token*.

En la tabla 3.8 se muestran las características que indican que un *token* pertenece a una clase con mayor seguridad. Puede apreciarse que las categorías de Wikipedia «Estados miembros de la ONU», «Nombres» y «Redes Sociales» son indicadores de entidades de tipo lugar, de tipo persona y misceláneas respectivamente. También se aprecia que los pronombres y signos de puntuación no suelen pertenecer a entidades. Tampoco suelen ser entidades los *tokens* con mapeo *aa*, es decir, los *tokens* escritos enteramente en minúsculas. Los *tokens* que tienen como descripción «ciudad» y «futbolista» son, generalmente, de tipo lugar y persona respectivamente.

En la tabla 3.7 puede verse que los sustantivos suelen representar entidades y que el primer *token* de un *tweet* no suele ser de la forma *I-X*. Los *tokens* que tienen asociada una página de Wikipedia suelen pertenecer a entidades.

Transiciones más probables	
B-PER	I-PER
B-MISC	I-MISC
I-MISC	I-MISC
I-LOC	I-LOC
B-LOC	I-LOC
B-ORG	I-ORG
I-PER	I-PER
I-ORG	I-ORG
O	O
B-LOC	B-LOC
O	B-PER
O	B-MISC
O	B-ORG
O	B-LOC
B-ORG	B-ORG

Tabla 3.6: Transiciones más probables

Transiciones menos probables	
O	I-MISC
O	I-LOC
O	I-ORG
O	I-PER
B-LOC	I-ORG
B-LOC	I-MISC
B-ORG	I-PER
B-ORG	I-LOC
B-PER	B-PER
B-PER	I-LOC
B-PER	I-MISC
I-PER	I-MISC
B-MISC	I-PER
I-LOC	I-PER
B-LOC	O

Tabla 3.7: Transiciones menos probables

Clase	**Característica**
O	t.first
O	t.shape: -
O	t.shape: aa
B-LOC	wiki_category: Estados miembros de la ONU
O	t.postag: puntuación
B-PER	wiki_category: Nombres
B-LOC	wiki_description: ciudad
O	t.postag: pronombre
B-LOC	t-1: en
B-MISC	wiki_category: Redes Sociales
B-LOC	wiki_description: capital
B-LOC	t-1: en
B-PER	wiki_category: Personas
I-PER	wiki_description: futbolista
I-PER	wiki_category: Apellidos

Tabla 3.8: Características que indican que un token pertenece a una clase con mayor seguridad

Clase	**Característica**
O	t.postag: sustantivo
O	wiki_category: Estados miembros de la ONU
O	wiki_description: capital
I-LOC	t.first
B-PER	t-1: en
O	wiki_category:Nombres
I-MISC	t.first
I-ORG	t.first
I-PER	t.first
B-PER	wiki_description: ciudad
B-ORG	t.shape: aa
O	t-1: calle
B-PER	t-1.shape: Aa
O	wiki_category: Nombres españoles
O	has_wiki_page

Tabla 3.9: Características que indican que un token no pertenece a una clase con mayor seguridad

3.2.3. Reducción de dimensiones

Con los recursos disponibles resulta imposible realizar el proceso de aprendizaje utilizando algoritmos que consumen mucha memoria como «k vecinos más cercanos (KNN)». Además, es posible que los algoritmos empleados no sean capaces de lidiar con conjuntos de entrenamiento más grandes o que las corridas sean muy demoradas. Por esto, se valora el impacto de reducir las dimensiones del problema con distintas variantes.

Las técnicas de reducción de dimensiones más utilizadas son Análisis de Componentes Principales (PCA)[52] y Descomposición en Valores Singulares Truncada (*Truncated SVD*)[45]. PCA busca la combinación de características que mejor refleja la varianza de los atributos originales. SVD es una factorización de una matriz con múltiples aplicaciones. Truncanted SVD es una variante que computa solo los k valores singulares más grandes. Ambos algoritmos buscan reducir el número de columnas, es decir, características; y preservar la similitud entre filas, en este caso: *tokens*.

Debido a las dimensiones del problema y la complejidad espacial de PCA, con las condiciones disponibles es imposible su utilización. A continuación se muestran los resultados obtenidos al utilizar *Truncated SVD* con distintos

valores de k antes de aplicar el clasificador PAC. Al usar *Truncated SVD* con 1000 componentes se obtiene una disminución de la medida F1 de apenas un 14 % respecto a la obtenida al aplicar PAC con las 118196 características.

No. de componentes	Precisión	Exhaustividad	Medida F1	%
5	0.027	0.032	0.029	6 %
10	0.084	0.102	0.092	20 %
50	0.266	0.173	0.210	46 %
100	0.322	0.222	0.262	57 %
500	0.438	0.323	0.372	81 %
1000	0.451	0.351	0.394	86 %

Tabla 3.10: Precisión, exhaustividad y medida F1 utilizando *Truncated SVD*

Subconjuntos de características

Otra manera de reducir las dimensiones del problema es utilizar un subconjunto de las características disponibles. Esta manera de reducir las dimensiones del problema permitirá, además, valorar la importancia de cada uno de los conjuntos de atributos utilizado. Estas características pueden dividirse en tres grupos: ortográficas, sintácticas y gramaticales (clásicas), no supervisadas y globales. A estas se suma una característica que indica si el uso de mayúsculas en un mensaje es el adecuado.

En la tabla 3.11 se muestran la precisión, exhaustividad y medida F1 obtenida utilizando distintos subconjuntos de atributos. Se muestra, también, el por ciento que representa la medida F1 del resultado obtenido utilizando todos los atributos (0.458). Puede observarse que los mejores resultados se obtienen al utilizar características clásicas y globales. Los tres conjuntos de características realizan la clasificación de entidades en nueve clases con una precisión superior a 0.250.

Al eliminar la categoría gramatical de la lista de atributos se obtiene una disminución de solo un 3 % respecto a la medida F1 original. Por lo que, en casos donde no se cuente con una anotación realizada manualmente, se puede prescindir de este atributo sin una gran penalización en los resultados.

Eliminación de stop words

Con la eliminación de los *stop words* y signos de puntuación se reduce a 73792 el número de *tokens*. Esto representa un 52 % del número de *tokens*

Características	**Precisión**	**Exhaustividad**	**Medida F1**	**%**
Clásicas	0.532	0.391	0.436	95 %
No supervisadas	0.250	0.155	0.169	37 %
Globales	0.462	0.270	0.329	72 %
Sin *postagging*	0.536	0.393	0.442	97 %

Tabla 3.11: Rendimiento utilizando distintos subconjuntos de características y PAC

original. El número de características se reduce a 1671541, un 57 % del número original. Los resultados son ligeramente superiores, como se muestra en la tabla 3.12, por lo que puede decirse que la eliminación de *stop words* no influye significativamente en el resultado de la clasificación.

Clasificador	**Precisión**	**Exhaustividad**	**Medida F1**	**%**
PAC	0.543	0.438	0.470	102 %
CRF	0.619	0.482	0.521	101 %

Tabla 3.12: Rendimiento obtenido tras eliminar los stop words

3.3. Self-training

El *self-training* depende de dos variables: el tamaño del conjunto de entrenamiento inicial (n) y el número de mensajes a añadir al conjunto de entrenamiento en cada iteración (m). En cada iteración se entrenan cinco clasificadores: PAC, SGD, Random Forest, CRF (L2-SGD) y CRF (L-BFGS). Los m mensajes más relevantes, según los criterios que se discuten a continuación, pasan a formar parte del conjunto de entrenamiento. Este proceso se repite hasta haber anotado todo el corpus.

Una primera variante para elegir los m mensajes a añadir al conjunto de entrenamiento consiste en medir la concordancia entre clasificadores. Con este fin, se utiliza una implementación personalizada de la medida *kappa de Fleiss*[37]. Se usa esta medida, en lugar de implementaciones disponibles de otros coeficientes más populares como el *kappa de Cohen*[23], pues estos solo permiten analizar la concordancia entre dos clasificaciones y en este caso se cuenta con cinco. Cada token es etiquetado con la clase asignada por un mayor número de clasificadores. En caso de empate entre una clase

que representa una entidad y una de tipo *O*, se elige la clase que representa una entidad. De esta forma se incentiva el descubrimiento de entidades. En caso de empate entre dos clases que representan entidades, se elige una de forma aleatoria. Esta variante exhibe una baja exhaustividad debido a que favorece la adición al corpus de *tweets* sin entidades.

La segunda variante busca aumentar la exhaustividad. Para cada *token* se toman, de las cinco clasificaciones obtenidas, las que representan entidades. Si la más común de estas clasificaciones aparece al menos dos veces, se asigna al *token*. De lo contrario, el *token* se etiqueta con la categoría *O*. Al corpus se añaden los *tweets* que tienen mayor promedio de *tokens* identificados como entidades. Esta segunda variante produce resultados significativamente superiores a la anterior.

Los resultados obtenidos utilizando la segunda variante de selección y para distintos valores de n y m se muestran en la tabla 3.13. Utilizando 2000 *tweets* se alcanzan resultados superiores a los obtenidos empleando PAC con todo el conjunto de entrenamiento. Los valores utilizados en la propuesta final para n y m son 1000 y 500 respectivamente. El uso de estos valores para el algoritmo de *self-training* requiere la anotación de 1000 *tweets* y conlleva una disminución de solo un 4 % respecto a la medida F1 obtenida utilizando PAC con todo el conjunto de entrenamiento.

n	m	Precisión	Exhaustividad	Medida F1
500	500	0.516	0.365	0.370
500	1000	0.562	0.338	0.389
1000	500	0.540	0.410	0.438
1000	1000	0.602	0.365	0.428
2000	500	0.585	0.453	**0.495**
2000	1000	0.600	0.408	0.468
3000	500	0.509	**0.476**	0.482
3000	1000	**0.617**	0.433	0.494

Tabla 3.13: Rendimiento utilizando distintos valores de n y m para el algoritmo de self-training

3.4. Comparación con otros sistemas

En la tabla 3.14 se muestra una comparación entre la propuesta final de este trabajo y otras metodologías existentes. *Stanford NER Tagger* es el

sistema de reconocimiento de entidades con mejores resultados reportados en dominios formales[29][84]. *T-NER* es uno de los sistemas con mejores resultados en *Twitter*[29].

En el caso del *Stanford NER Tagger* se muestran los resultados obtenidos utilizando su modelo para reconocimiento de entidades en español y utilizando un modelo entrenado con 1000 *tweets* del *xLiMe Twitter Corpus*. *T-NER* no puede ser entrenado con un nuevo corpus. Se observa una mejoría de un 10 % respecto al *Stanford NER Tagger* entrenado, de un 21 % respecto al *Stanford NER Tagger* para el idioma español y de un 49 % respecto a *T-NER* en cuanto a medida F1.

Sistema	Precisión	Exhaustividad	Medida F1
Stanford NER	0.299	**0.456**	0.361
Stanford NER (entrenado)	**0.627**	0.292	0.398
T-NER	0.319	0.281	0.293
Propuesta	0.602	0.408	**0.438**

Tabla 3.14: Comparación de sistemas de reconocimiento de entidades en *xLiMe Twitter Corpus*

3.5. Comportamiento en otro dominio

Para comprobar la portabilidad de la propuesta, a continuación se muestran los resultados obtenidos por la propuesta en el corpus de la Conferencia de Aprendizaje Computacional del Lenguaje Natural (CoNLL 2003)[86]. Este corpus está compuesto por cables noticiosos de la agencia Reuters[2]. El corpus contiene 11755 oraciones. Para utilizar la metodología desarrollada, se considera cada oración como un *tweet*.

En este caso, no es necesario un corpus formal para determinar si el uso de mayúsculas y minúsculas es correcto, pues los textos son extraídos de cables noticiosos. Como corpus de datos no anotados, para la obtención de los atributos no supervisados, se utiliza una porción del *20 Newsgroups Corpus*[60]. Como fuente de conocimiento externo se utiliza la Wikipedia en inglés. En las tablas 3.15 y 3.16 se muestran los resultados obtenidos.

[2] `http://www.reuters.com/`

Sistema	Precisión	Exhaustividad	Medida F1
Propuesta	0.718	0.729	0.723

Tabla 3.15: Rendimiento de la propuesta en el corpus de CoNLL

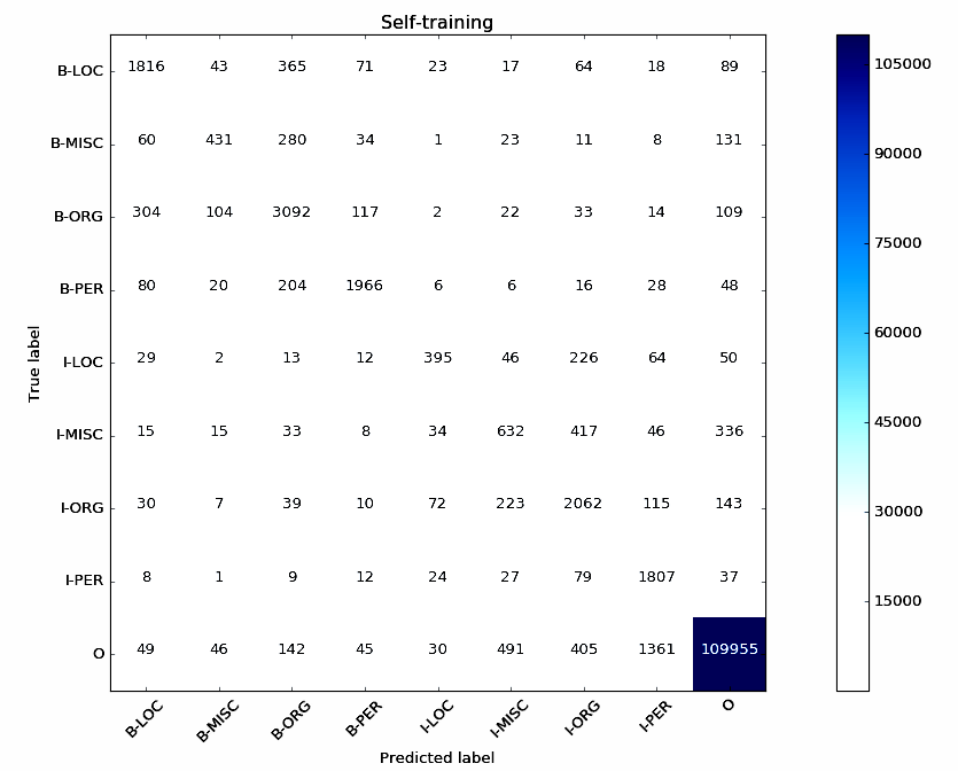

Figura 3.3: Matriz de confusión obtenida utilizando la propuesta final en el corpus CoNLL

Tipo de Entidad	**Precisión**	**Exhaustividad**	**Medida F1**
B-LOC	0.760	0.725	0.742
B-MISC	0.646	0.441	0.524
B-ORG	0.741	0.815	0.776
B-PER	**0.864**	0.828	**0.846**
I-LOC	0.677	0.427	0.524
I-MISC	0.526	0.413	0.463
I-ORG	0.644	0.764	0.699
I-PER	0.747	**0.902**	0.817
O	**0.991**	**0.984**	**0.987**

Tabla 3.16: Precisión, exhaustividad y medida F1 por tipo de entidad obtenidos utilizando self-training en el corpus CoNLL

Conclusiones

En la tesis se estudia el proceso de Reconocimiento de Entidades Nombradas y se propone una metodología para el reconocimiento de entidades en mensajes cortos. Para esto, se emplean características que pueden dividirse en tres grupos: ortográficas, sintácticas y gramaticales (clásicas), no supervisadas y globales. A estas se suma una característica que indica si el uso de mayúsculas en un mensaje es el adecuado. El proceso de clasificación se realiza utilizando *self-training* con PAC, SGD, Random Forest, CRF (L2-SGD) y CRF (L-BFGS). Estos clasificadores son los que mejores resultados presentan entre los 15 clasificadores valorados. La propuesta final presenta una *medida F1* de 0,438.

Al comparar la propuesta presentada con otras metodologías existentes se observa una mejoría de un 10 % respecto al *Stanford NER Tagger*, el algoritmo con mejores resultados reportados en la bibliografía para corpus formales y de un 49 % respecto a *T-NER*, uno de los dos sistemas para el reconocimiento de entidades en *Twitter* con mejores resultados reportados. La propuesta que se presenta tiene, además, la ventaja de necesitar solo un pequeño número de mensajes anotados para su entrenamiento. Igualmente es una propuesta portable a otros dominios como se demuestra con su medida F1 de 0.723 en un corpus formal como el de CONLL.

Recomendaciones

Para futuros trabajos se propone un estudio a fondo de la elección de clasificadores para el proceso de *self-training* y de los valores adecuados para los parámetros que regulan el tamaño del conjunto inicial y el número de mensajes a añadir al conjunto de entrenamiento en cada iteración. Debe estudiarse, también, el algoritmo de selección de mensajes a añadir, utilizando otras medidas de concordancia y teniendo en cuenta la probabilidad de correctitud de la clasificación facilitada por cada algoritmo.

Otra forma de mejorar los resultados obtenidos con aprendizaje semisupervisado puede ser cambiar el algoritmo utilizado. En particular, se recomienda el uso de *co-training* en lugar de *self-training* como técnica de aprendizaje.

Se recomienda, además, aplicar la metodología en otros dominios para hacer un análisis más completo de su portabilidad. Para esto puede valorarse el uso de otras *wikis*. Puede, por ejemplo, aplicarse la metodología a un conjunto de mensajes enviados desde Cuba y utilizarse la Enciclopedia Cubana en la Red (EcuRed)[3] como fuente de conocimiento externo.

Otra recomendación es evaluar el impacto de enriquecer el conjunto de mensajes cortos anotados con oraciones tomadas de textos estructurados anotados. Los corpus de textos formales como cables noticiosos anotados son más abundantes que los de mensajes cortos anotados que son muy escasos.

[3] `https://www.ecured.cu/EcuRed:Enciclopedia_cubana`

Anexos

Anexo 1: Características utilizadas en la clasificación

Característica	Descripción
t_0	token actual
t_{-1}	token precedente
t_{+1}	token siguiente
$t_0.lower$	token en minúsculas
$t_0[:1]$, $t_0[:2]$, $t_0[:3]$	prefijos de tamaño uno, dos y tres
$t_0[-1:]$, $t_0[-2:]$, $t_0[-3:]$	sufijos de tamaño uno, dos y tres
$t_0.isupper$	indica si el token está compuesto solo por mayúsculas
$t_0.istitle$	indica si el token comienza con una mayúscula
$t_0.isdigit$	indica si el token está compuesto únicamente por dígitos
$t_0.shape$	mapeo de un token a una expresión que representa el uso de mayúsculas y minúsculas en el token
$t_{-1}.shape, t_{+1}.shape$	
$t_0.postag$	categoría gramatical del token
$t_{-1}.postag, t_{+1}.postag$	
$t_{-1}\&t_0$	concatenación del token precedente y el actual
$t_0\&t_1$	concatenación del token actual y el siguiente
$t_0.first$	indica si el token es el primero del tweet
$t_0.last$	indica si el token es el último del tweet

$t_0.brown-7$	clúster de Brown al que pertenece el token (100 clústers)
$t_0.brown-9$	clúster de Brown al que pertenece el token (80 clústers)
$t_0.brown-11$	clúster de Brown al que pertenece el token (50 clústers)
$t_0.clark$	clúster de Clark al que pertenece el token (100 clústers)
$t_0.word2vec$	clúster de Word2vec al que pertenece el token (100 clústers)
$t_0.has\_wiki\_page$	indica si existe una página de Wikipedia cuyo título contenga al token
$t_0.wiki\_page$	la página de Wikipedia más relevante en el contexto del token cuyo título contenga al token
$t_0.wiki\_category$	categoría de $t_0.wiki\_page$ más frecuente en el corpus
$t_0.wiki\_description$	primer sustantivo que sucede a la primera ocurrencia de una forma del verbo ser en $t_0.wiki\_page$
$proper\_case\_use$	indica si el uso de mayúsculas y minúsculas en el tweet al que pertenece el token es adecuado

Tabla 3.17: Lista completa de atributos utilizados en la clasificación

Anexo 2: Categorías Gramaticales de los tokens

Tag	Alemán	Italiano	Español
Adjetivo	2514	7684	5741
Preposición	4333	14960	13467
Adverbio	4173	8476	6116
Conjunción	1576	6737	6684
Determinante	2990	9811	10037
Interjección	225	1427	1109
Sustantivo	11057	30759	23230
Número	1176	2550	1568

Pronombre	4530	7737	10333
Puntuación	8650	20529	14102
Forma verbal	6506	21793	19640
Otro	1936	1503	3033
Continuación	918	4227	3422
Emoticono	449	1076	951
Hashtag	1895	3035	1805
Mención	1984	6519	9070
URL	1923	4494	3019

Tabla 3.18: Categorías gramaticales en el *xLiMe Twitter Corpus*

Glosario

API

API Application Programming Interface. Conjunto de comandos, funciones, protocolos y objetos que los programadores pueden utilizar para crear software o interactuar con un sistema externo. Provee a los desarrolladores de comandos estándares para realizar operaciones comunes, evitando escribir todo el código nuevamente. 32

cluster

conjunto de objetos similares y correlacionados. 11, 21, 22, 32, 33

clustering

técnica para particionar un conjunto en subconjuntos de objetos similares y correlacionados. 21

corpus

es un conjunto de datos o textos de un mismo tipo que sirve de base a una investigación. 3, 6, 8, 10, 12, 15, 21, 24, 25, 30

dendograma

es una representación gráfica o diagrama de datos en forma de árbol que organiza los datos en subcategorías que se van dividiendo en otras hasta llegar al nivel de detalle deseado. 21

emoticono

es una secuencia de caracteres ASCII que expresan una emoción. Se emplean frecuentemente en mensajes de correo electrónico, foros, SMS y chats. 8, 16, 17, 32, 48

exhaustividad

es una métrica empleada para medir el rendimiento de los sistemas de recuperación de información y reconocimiento de patrones. Es la fracción de instancias relevantes que han sido recuperadas. 6, 9, 33, 34, 39, 41

hashtag

es una cadena de caracteres formada por una o varias palabras concatenadas y precedidas por un numeral (#) con el fin de que tanto el sistema como el usuario la identifiquen de forma rápida. 8, 17, 48

línea de base

es un algoritmo que predice aleatoriamente o utilizando simples estadísticas. Se emplea para valorar la mejora de un algoritmo con respecto a una solución sencilla. 21, 34

medida F1

La medida F1 es una medida de la exactitud de un modelo. Es dos veces la media armónica entre la precisión y la exhaustividad del modelo. 34, 35, 39–42, 44

microblogging

es un servicio que permite a sus usuarios enviar y publicar mensajes breves, generalmente solo de texto. 1, 14, 17

precisión

es una métrica empleada para medir el rendimiento de los sistemas de recuperación de información y reconocimiento de patrones. Es la fracción de instancias recuperadas que son relevantes. 6, 33, 34, 39

stop word

son palabras que suelen eliminarse antes o después de un procesamiento de lenguaje natural por su alta frecuencia de aparición en textos y su bajo aporte semántico. 18, 40

token

es una palabra u otra unidad atómica de un texto. 9–11, 13, 15, 16, 18, 19, 21–23, 25, 28, 30, 32–34, 36, 37, 39–41

tweet

(o tuit) es un mensaje enviado vía Twitter. IV, VI, 8, 16–18, 21–25, 30, 32, 37, 41, 42

Twitter

(`http://www.twitter.com`) es una red social de microblogging que permite a sus usuarios enviar mensajes de texto plano con un máximo de 140 caracteres, llamados tweets. IV, 1–3, 8, 14, 17, 32, 41, 44

wiki

es el nombre que recibe un sitio web cuyas páginas pueden ser editadas directamente desde el navegador, donde los usuarios crean, modifican o eliminan contenidos. VI, 3, 15, 45

Wikipedia

(`https://www.wikipedia.org`) es una enciclopedia libre, políglota y editada colaborativamente. Cuenta con más de 37 millones de artículos en 287 idiomas. 10, 14, 22, 23, 36, 37, 43

WordNet

es una base de datos léxica del Idioma Inglés. Agrupa palabras en inglés en conjuntos de sinónimos llamados synsets, proporcionando definiciones cortas y generales, y almacena las relaciones semánticas entre los conjuntos de sinónimos. 11, 12

Bibliografía

[1] Abelleira, Alicia Pérez y Alejandra Carolina Cardoso: *Técnicas de extracción de entidades con nombre.* En *14th Argentine Symposium on Artificial Intelligence*, páginas 109–120, 2013. (Citado en la página 2).

[2] Alfonseca, Enrique y Suresh Manandhar: *An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery.* 2002. (Citado en la página 6).

[3] Almeida-Cruz, Yudivián, Suilan Estévez-Velarde y Alejandro Piad-Morffis: *Detección de Idioma en Twitter.* Revista Internacional de Gestión del Conocimiento y la Tecnología., 2014. (Citado en la página 2).

[4] Ando, Rie Kubota y Tong Zhang: *A HighPerformance Semi-Supervised Learning Method for Text Chunking.* En *Proceedings of the 43rd Annual Meeting of ACL*, páginas 1–9, 2005. (Citado en la página 21).

[5] Asahara, Masayuki y Yuji Matsumoto: *Japanese Named Entity Extraction with Redundant Morphological Analysis.* En *Proceedings Human Language Technology conference - North American chapter of the Association for Computational Linguistics*, 2003. (Citado en la página 11).

[6] Bengfort, Benjamin: *A Survey of Stochastic and Gazetteer Based Approaches for Named Entity Recognition.* 2012. (Citado en las páginas 6, 9 y 11).

[7] Berger, Adam L., Vincent J. Della Pietra y Stephen A. Della Pietra: *A Maximum Entropy Approach to Natural Language Processing.* Comput. Linguist., 22(1):39–71, Marzo 1996, ISSN 0891-2017. (Citado en la página 28).

[8] Bikel, Daniel M., Scott Miller, Richard Schwartz y Ralph Weischedel: *Nymble: a High-Performance Learning Name-finder*. En *Proceedings Conference on Applied Natural Language Processing.*, 1997. (Citado en la página 11).

[9] Bikel, Daniel M., Richard Schwartz y Ralph M. Weischedel: *An algorithm that learns whats in a name*. Machine Learning, 34(1-3):211–231, 1999. (Citado en la página 28).

[10] Bird, Steven, Ewan Klein y Edward Loper: *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edición, 2009, ISBN 0596516495, 9780596516499. (Citado en la página 34).

[11] Blum, Avrim y Tom Mitchell: *Combining Labeled and Unlabeled Data with Co-training*. En *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, páginas 92–100, New York, NY, USA, 1998. ACM, ISBN 1-58113-057-0. (Citado en la página 12).

[12] Bontcheva, Kalina, Leon, Mark A. Groenwood Adam Funk, Diana Maynard y Niraj Aswani: *TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text.* En *Proceedings of the International Conference on Recent Advances in Natural Language Processing.* Association for Computational Linguistics, 2013. (Citado en la página 16).

[13] Borthwick, Andrew, John Sterling, Eugene Agichtein y Ralph Grishman: *NYU: Description of the MENE named entity system as used in MUC-7*. En *Proceedings of the Seventh Message Understanding Conference (MUC-7*, 1998. (Citado en la página 11).

[14] Breiman, Leo: *Random Forests*. Mach. Learn., 45(1):5–32, Octubre 2001, ISSN 0885-6125. (Citado en la página 28).

[15] Breiman, Leo, J. H. Friedman, R. A. Olshen y C. J. Stone: *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984. (Citado en la página 26).

[16] Brown, Peter F., Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra y Jenifer C. Lai: *Class-Based n-gram Models of Natural Language*. 1992. (Citado en las páginas 21 y 32).

[17] Burges, Christopher J. C.: *A Tutorial on Support Vector Machines for Pattern Recognition.* Data Min. Knowl. Discov., 2(2):121–167, Junio 1998, ISSN 1384-5810. (Citado en la página 26).

[18] Carreras, Xavier, Lluís Márquez y Lluís Padró: *A Simple Named Entity Extractor using AdaBoost.* En *Proceedings in CONLL '03 Proceedings of the seventh conference on Natural language learning*, volumen 4, páginas 52–55. HLT-NAACL 2003, 2003. (Citado en las páginas 11 y 19).

[19] Chapelle, Olivier, Bernhard Schlkopf y Alexander Zien: *Semi-Supervised Learning.* The MIT Press, 1st edición, 2010, ISBN 0262514125, 9780262514125. (Citado en la página 12).

[20] Charniak, Eugene, Curtis Hendrickson, Neil Jacobson y Mike Perkowitz: *Equations for part-of-speech tagging.* AAAI, páginas 784–789, 1993. (Citado en la página 21).

[21] Chiticariu, Laura, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss y Shivakumar Vaithyanathan: *Domain Adaption of Rule-Based Annotators for Named Entity Recognition Tasks.* En *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, páginas 1002–1012. EMNLP '10, 2010. (Citado en la página 9).

[22] Clark, Alexander: *Combining distributional and morphological information for part of speech induction.* En *In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, volumen 1, páginas 59–66. Association for Computational Linguistics, 2003. (Citado en las páginas 22 y 32).

[23] Cohen, Jacob: *Weighted Kappa - Nominal Scale Agreement with Provision for Scaled Disagreement Or Partial Credit.* Psychological Bulletin, 1968. (Citado en la página 40).

[24] Collins, Michael: *Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms.* En *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, páginas 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. (Citado en la página 34).

[25] Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz y Yoram Singer: *Online Passive-Aggressive Algorithms.* J. Mach. Learn.

Res., 7:551–585, Diciembre 2006, ISSN 1532-4435. (Citado en las páginas 27 y 34).

[26] crfsuite sklearn: *sklearn-crfsuite*, Mayo 2017. `http://sklearn-crfsuite.readthedocs.io/en/latest/`. (Citado en la página 34).

[27] Cucchiarelli, Alessandro y P Velardi P.: *Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence*. Computational Linguistics, (27):123–131, 2001. (Citado en la página 13).

[28] Derczynski, Leon, Kalina Bontcheva y Ian Roberts: *Broad Twitter Corpus: A Diverse Named Entity Recognition Resource*. 2016. (Citado en la página 8).

[29] Derczynski, Leon, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak y Kalina Bontcheva: *Analysis of named entity recognition and linking for tweets*. Information Processing and Management, 2015. (Citado en las páginas 8, 18, 34 y 42).

[30] Downey, Doug, Matthew Broadhead y Oren Etzioni: *Locating complex named entities in web text.* En *Proceedings of the 20th international joint conference on Artifical intelligence.*, 2007. (Citado en la página 8).

[31] Eddy, Sean R.: *Hidden markov models*. Current opinion in structural biology, 6(3):361–365, 1996. (Citado en la página 28).

[32] Erp, Marieke van, Giuseppe Rizzo y Raphaël Troncy: *Learning with the Web: Spotting Named Entities on the intersection of NERD and Machine Learning*. En *In Proceedings of 3rd International Workshop on Making Sense of Microposts*, 2013. (Citado en la página 8).

[33] Estévez-Velarde, Suilan y Yudivián Almeida-Cruz: *Minería de Opinión en Twitter: una aproximación desde el aprendizaje supervisado.* Tesis de Licenciatura, Facultad de Matemática y Computación, Universidad de La Habana. 2015. (Citado en las páginas 2 y 18).

[34] Evans, Richard: *A Framework for Named Entity Recognition in the Open Domain*. En *Proceedings of Recent Advances in Natural Language Processing*, 2003. (Citado en la página 12).

[35] Feldman, Ronen y James Sanger: *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, NY, USA, 2006, ISBN 0521836573, 9780521836579. (Citado en la página 16).

[36] Finkel, Jenny Rose, Trond Grenager y Christopher D. Manning: *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. En *Proceedings of ACL*, páginas 363–370, 2005. (Citado en la página 7).

[37] Fleiss, J.L. *y cols.*: *Measuring nominal scale agreement among many raters*. Psychological Bulletin, 76(5):378–382, 1971. (Citado en la página 40).

[38] Florian, Radu: *Named entity recognition as a house of cards: classifier stacking*. En *Proceedings of the 6th conference on Natural language learning.*, volumen 20. COLING, 2002. (Citado en la página 8).

[39] Freund, Yoav y Robert E Schapire: *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. J. Comput. Syst. Sci., 55(1):119–139, Agosto 1997, ISSN 0022-0000. (Citado en la página 28).

[40] Freund, Yoav y Robert E. Schapire: *Large Margin Classification Using the Perceptron Algorithm*. En *Machine Learning*, páginas 277–296, 1998. (Citado en la página 27).

[41] Grishman, Ralph y Beth Sundheim: *Message Understanding Conference-6: A brief history*. En *Proceedings in COLING '96 Proceedings of the 16th conference on Computational linguistics*, volumen 1, páginas 466–471, 1996. (Citado en las páginas 5 y 6).

[42] Group, The Stanford Natural Language Processing: *Stanford Named Entity Recognizer (NER)*, 2017. `https://nlp.stanford.edu/software/CRF-NER.shtml`. (Citado en las páginas 9 y 21).

[43] Gutiérrez, Raúl, Andrés Castillo, Víctor Bucheli y Oswaldo Solarte: *Reconocimiento de entidades nombradas para el idioma Español y su aplicación en la vigilancia tecnológica* . Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software., 2015. (Citado en la página 6).

[44] Habib, Mena B. y Maurice van Keulen: *Unsupervised Improvement of Named Entity Extraction in Short Informal Context Using Disambiguation Clues*. 2011. (Citado en la página 8).

[45] Halko, N., P. G. Martinsson y J. A. Tropp: *Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions*. SIAM Rev., 53(2):217–288, Mayo 2011, ISSN 0036-1445. (Citado en la página 38).

[46] Hanisch, Daniel, Katrin Fundel, Mevissen Heinz-Theodor, Ralf Zimmer y Juliane Fluck: *ProMiner: rule-based protein and gene entity recognition*. BMC Bioinformatics, 6(1):S14, 2005, ISSN 1471-2105. (Citado en la página 6).

[47] Hastie, Trevor, Robert Tibshirani y Jerome Friedman: *The Elements of Statistical Learning: data mining, inference and prediction*. Springer, segunda edición, 2009. (Citado en la página 30).

[48] Hernández-Amador, Ariel y Yudivián Almeida-Cruz: *Influencia en Twitter: una propuesta basada en agrupamiento*. 2011. (Citado en la página 2).

[49] hmmlearn: *hmmlearn*, Mayo 2017. `http://hmmlearn.readthedocs.io/en/stable/`. (Citado en la página 34).

[50] Inc., Statista: *Statistics and facts about Twitter*, 2017. `https://www.statista.com/topics/737/twitter/`. (Citado en la página 1).

[51] Jaynes, E. T.: *Information Theory and Statistical Mechanics*. Phys. Rev., 106:620–630, May 1957. (Citado en la página 28).

[52] Jolliffe, I.T.: *Principal Component Analysis*. Springer Verlag, 1986. (Citado en la página 38).

[53] Kazama, Jun'ichi y Kentaro Torisawa: *Exploiting Wikipedia as External Knowledge for Named Entity Recognition*. En *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, páginas 698–707, 2007. (Citado en las páginas 10 y 22).

[54] Kedzie, Chris, Kathleen McKeown y Fernando Diaz: *Predicting salient updates for disaster summarization*. En *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics.*, volumen 1,

páginas 1608–1617. Association for Computational Linguistics., 2015. (Citado en la página 2).

[55] Kim, Jin Dong, Tomoko Ohta, Yoshimasa Tsuruoka Yuka Tateisi y Nigel Collier: *Introduction to the Bio-entity Recognition Task at JNLPBA*. En *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, JNLPBA '04, páginas 70–75, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. (Citado en la página 6).

[56] Kitoogo, Fredrick Edward y Venansius Baryamureeba: *A Methodology for Feature Selection in Named Entity Recognition*. 2006. (Citado en la página 19).

[57] Kripke, Saul: *Naming and Necessity*. Harvard University Press, 1980. (Citado en la página 5).

[58] Krogel, Mark A. y Tobias Scheffer: *Multi-Relational Learning, Text Mining, and Semi-Supervised Learning for Functional Genomics*. Machine Learning, 57:61–81, 2004. (Citado en la página 13).

[59] Lafferty, John, Andrew McCallum, Fernando Pereira *y cols.*: *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. En *Proceedings of the eighteenth international conference on machine learning, ICML*, volumen 1, páginas 282–289, 2001. (Citado en la página 29).

[60] Lang, Ken: *20 newsgroups data set*, 1999. `http://www.ai.mit.edu/people/jrennie/20Newsgroups/`. (Citado en la página 42).

[61] Leaman, Robert, Chih Hsuan Wei y Zhiyong Lu: *tmChem: a high performance approach for chemical named entity recognition and normalization*. Journal of Cheminformatics, 7(1):S3, 2015, ISSN 1758-2946. (Citado en la página 6).

[62] Levy, Omer, Yoav Goldberg y Ido Dagan: *Improving Distributional Similarity with Lessons Learned from Word Embeddings*. Transactions of the Association for Computational Linguistics, 2015. (Citado en las páginas 22 y 32).

[63] Li, Chenliang, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun y Bu Sung Lee: *TwiNER: Named Entity Recognition in Targeted Twitter Stream*. En *Proceedings of the 35th International*

ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, páginas 721–730, New York, NY, USA, 2012. ACM, ISBN 978-1-4503-1472-5. (Citado en las páginas 8 y 17).

[64] Mascaro, Christopher y Sean Patrick Goggins: *Twitter as virtual town square: Citizen engagement during a nationally televised republican primary debate.* APSA 2012 Annual Meeting Paper, 2012. (Citado en la página 2).

[65] McCallum, Andrew y Wei Li: *Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons.* En *Proceedings Conference on Computational Natural Language Learning*, 2003. (Citado en la página 11).

[66] McCullagh, P. y J. A. Nelder: *Generalized Linear Models.* Chapman & Hall / CRC, London, 1989. (Citado en la página 11).

[67] Mederos, Oscar y Ariel Hernández-Amador: *Detección de tópicos en Twitter.* 2013. (Citado en la página 2).

[68] Mejer, Avihai y Koby Crammer: *Confidence in Structured-prediction Using Confidence-weighted Models.* En *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, páginas 971–981, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. (Citado en la página 34).

[69] Mikolov, Tomas, Kai Chen, Greg Corrado y Jeffrey Dean: *Efficient Estimation of Word Representations in Vector Space.* 2013. (Citado en las páginas 22 y 32).

[70] Mostafa, Mohamed M.: *More than words: Social networks text mining for consumer brand sentiments.* Expert Systems with Applications., 2013. (Citado en la página 2).

[71] Nadeau, David y Satoshi Sekine: *A survey of named entity recognition and classification.* Journal of Linguistical Investigationes, 2007. (Citado en las páginas 5, 6, 9, 10 y 11).

[72] Neubig, Graham, Yuichiroh Matsubayashi, Masato Hagiwara y Koji Murakami: *Safety information mining - what can NLP do in disaster.* En *Proceedings of 5th International Joint Conference on Natural Language Processing.*, páginas 965–973. Asian Federation of Natural Language Processing, 2011. (Citado en la página 2).

[73] Nothman, Joel, Nicky Ringland, Will Radford, Tara Murphy y James R. Curran: *Learning multilingual named entity recognition from Wikipedia*. Artificial Intelligence, páginas 151–175, 2013. (Citado en la página 14).

[74] Okazaki, Naoaki: *CRFsuite: a fast implementation of Conditional Random Fields (CRFs)*, 2007. `http://www.chokkan.org/software/crfsuite/`. (Citado en la página 34).

[75] O'Connor, Brendan: *Twokenizer*, 2014. `https://github.com/brendano/ark-tweet-nlp/blob/master/src/cmu/arktweetnlp/Twokenize.java`. (Citado en la página 16).

[76] O'Connor, Brendan, Michel Krieger y David Ahn: *TweetMotif: Exploratory Search and Topic Summarization for Twitter*. En *In Proceedings of the 4th International Conference on Weblogs and Social Media*, páginas 384–385, 2010. (Citado en la página 16).

[77] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay: *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12:2825–2830, 2011. (Citado en la página 33).

[78] Poibeau, Thierry y L Kosseim: *Proper Name Extraction from Non-Journalistic Texts*. En *Proceedings Computational Linguistics in the Netherlands*, 2001. (Citado en la página 6).

[79] Ramage, Daniel, David Hall, Ramesh Nallapati y Christopher D. Manning: *Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora*. En *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, páginas 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics, ISBN 978-1-932432-59-6. (Citado en la página 8).

[80] Ratinov, L y D Roth: *Design Challenges and Misconceptions in Named Entity Recognition*. En *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, páginas 147–155, 2009. (Citado en las páginas 7 y 21).

[81] Reese, Samuel, Gemma Boleda, Montse Cuadros, Lluís Padró y German Rigau: *Wikicorpus: A Word-Sense Disambiguated Multilingual*

Wikipedia Corpus. En *In Proceedings of 7th Language Resources and Evaluation Conference*, 2010. (Citado en la página 32).

[82] Rei, Luis, Dunja Mladenić y Simon Krek: *A Multilingual Social Media Linguistic Corpus*. 2014. (Citado en la página 30).

[83] Richman, Alexander E. y Patrick Schone: *Mining Wiki Resources for Multilingual Named Entity Recognition*. En *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies*, páginas 1–9, 2008. (Citado en la página 10).

[84] Ritter, Alan, Sam Clark y Oren Etzioni: *Named entity recognition in tweets: An experimental study.* En *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, 2011. (Citado en las páginas 2, 8, 17, 21, 23 y 42).

[85] Rowe, Matthew, Milan Stankovic, Aba Sah Dadzie, B.P Nunes y Amparo Elizabeth Cano: *Making sense of microposts (#msm2013): Big things come in small packages.* En *Proceedings of the WWW Conference - Workshops*, 2013. (Citado en la página 17).

[86] Sang, Erik F. Tjong Kim y Fien De Meulder: *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition.* En *Proceedings of CoNLL-2003*, 2003. (Citado en las páginas 6, 11 y 42).

[87] Saragawi, Sunita: *Information Extraction.* Foundations and Trends in Databases, página 261–377, 2007. (Citado en la página 6).

[88] Sekine, Satoshi: *Description of the Japanese NE System Used For Met-2.* En *Proceedings Message Understanding Conference.*, 1998. (Citado en la página 11).

[89] Sekine, Satoshi y Chikashi Nobata: *Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy.* En *Proceedings Conference on Language Resources and Evaluation*, 2004. (Citado en la página 6).

[90] Shalev-Shwartz, Shai, Yoram Singer y Nathan Srebro: *Pegasos: Primal Estimated sub-GrAdient SOlver for SVM.* En *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, páginas 807–814, New York, NY, USA, 2007. ACM, ISBN 978-1-59593-793-3. (Citado en la página 34).

[91] Shen, Dan, Jie Zhang, Jian Su, GuoDong Zhou y ChewLim Tan: *Multi-Criteria-based Active Learning for Named Entity Recognition.* 2004. (Citado en la página 19).

[92] Sullivan, Dan: *Document Warehousing and Text Mining.* John Wiley & Sons, Inc., 2001. (Citado en la página 5).

[93] Suzuki, Jun y Hideki Isozaki: *Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data.* En *In ACL*, páginas 665–673, 2008. (Citado en la página 21).

[94] Taylor, Ann, Mitchell Marcus y Beatrice Santorini: *The Penn Treebank: An Overview*, 2003. (Citado en la página 16).

[95] Tkachenko, Maksim y Andrey Simanovsky: *Named Entity Recognition: Exploring Features.* En *Proceedings of KONVENS 2012 (Main track: oral presentations)*, 2012. (Citado en las páginas 19 y 21).

[96] Toutanova, Kristina, Dan Klein, Christopher D. Manning y Yoram Singer: *Feature-rich part-of-speech tagging with a cyclic dependency network.* En *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology.*, volumen 1. NAACL, 2003. (Citado en la página 21).

[97] Tumasjan, Andranik, Timm O. Sprenger, Philipp G. Sandner y Isabell M. Welpe: *Predicting Elections with Twitter. What 140 characters reveal about political sentiment.* En *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, páginas 178–185, 2010. (Citado en la página 2).

[98] Turian, Joseph, Yoshua Bengio y Lev Ratinov: *Word representations: A simple and general method for semi-supervised learning.* 2010. (Citado en las páginas 12 y 21).

[99] Weinberger, Kilian, Anirban Dasgupta, John Langford, Alex Smola y Josh Attenberg: *Feature Hashing for Large Scale Multitask Learning.* En *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, páginas 1113–1120, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-516-1. (Citado en la página 32).

[100] Wikimedia: *Estadísticas de Wikipedia Español*, Enero 2017. `https://stats.wikimedia.org/ES/TablesWikipediaES.htm`. (Citado en la página 14).

[101] Zhang, Harry: *The Optimality of Naive Bayes.* En Barr, Valerie y Zdravko Markov (editores): *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004).* AAAI Press, 2004. (Citado en la página 26).

[102] Zhang, Tong: *Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms.* En *ICML 2004: Proceedings of the 21st International Conferencde on Machine Learning*, páginas 919–926, 2004. (Citado en la página 27).

[103] Zhou, GuoDong, Dan Shen, Jie Zhang, Jian Su y ChewLim Tan: *Recognition of Protein/Gene Names from Text using an Ensemble of Classifiers. BMC Bioinformatics.* 2004. (Citado en la página 19).