Facultad de Ciencias y Tecnologías Computacionales

Extracción de entidades nombradas en artículos de prensa en español.

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Karel Aguilera Murrell

Tutor(es): P.A., Ing. Yusniel Hidalgo Delgado, Dr. C.

La Habana, noviembre de 2022

Año 64 de la Revolución

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título **“Extracción de entidades nombradas en artículos de prensa en español”** concede(n) a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara(n) como único(s) autores de su contenido. Para que así conste firma(n) la presente a los <día> días del mes de <mes> del año <año>.

<nombre del autor>	**<nombre del autor>**
_____	_____
Firma del Autor	Firma del Autor
<nombre del tutor>	**<nombre del tutor>**
_____	_____
Firma del Tutor	Firma del Tutor

DATOS DE CONTACTO

Yusniel Hidalgo Delgado Profesor Auxiliar en el Departamento de Informática de la Facultad de Ciencias y Tecnologías Computacionales de la Universidad de las Ciencias Informáticas (UCI). Obtuvo el título de Ingeniero en Ciencias Informáticas, Máster en Informática Aplicada y Doctor en Informática por la UCI en el 2010, 2015 y 2021 respectivamente. Sus intereses de investigación son Web Semántica, Datos Enlazados e Ingeniería Ontológica. Es miembro de la Asociación Cubana de Reconocimiento de Patrones y de la Sociedad Cubana de Matemática y Computación. Posee varias publicaciones científicas en revistas y congresos internacionales. Actualmente se desempeña como Jefe del Grupo de investigación de Web Semántica y Jefe del Departamento de Informática de la CITEC de dicha universidad.

Opinión del(os) tutor(es)

RESUMEN

La prensa digital es un medio que mezcla en una misma plataforma diversas clases de formatos, como resultado, la demanda de cierto tipo de servicios por parte de la industria, exigen procesos más económicos y eficientes de obtener los resultados. El bajo desempeño de los modelos multilenguaje es un problema que afecta la calidad de los sistemas basados en aprendizaje automático para el procesamiento del lenguaje natural usados en la clasificación de entidades para enriquecer los servicios de la prensa digital, esto conlleva a la siguiente interrogante: ¿cómo extraer entidades nombradas de noticias en español de forma automática de varios dominios? En esta investigación se propone un método para la extracción de entidades nombradas en artículos de prensa en español basado en etapas que siguen un modelo codificador-decodificador , donde la salida de una fase constituye la entrad a de la próxima. Luego de realizado todo el proceso de desarrollo y la fase de prueba, se obtuvo como resultado un componente de software completamente funcional y fácil de manejar.

PALABRAS CLAVE

Noticia, entidad, aprendizaje profundo, clasificación

ABSTRACT

The digital press is a medium that mixes in the same platform different kinds of formats, as a result, the demand for certain types of services by the industry, demand cheaper and more eff icient processes to obtain the results. The poor performance of multi-language models is a problem that affects the quality of machine learning-based systems for natural language processing used in the classification of entities to enrich the services of the digital press, this leads to the following question: how to extract named entities of news in Spanish automatically from various domains? This research proposes a method for the extraction of entities named in press articles in Spanish based on stages that follow an encoder-decoder model, where the output of one phase constitutes the input of the next. After completing the entire development process and testing phase, a fully functional and easy-to-use software component was obtained.

KEYWORDS

News, entity, deep learning, classification

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

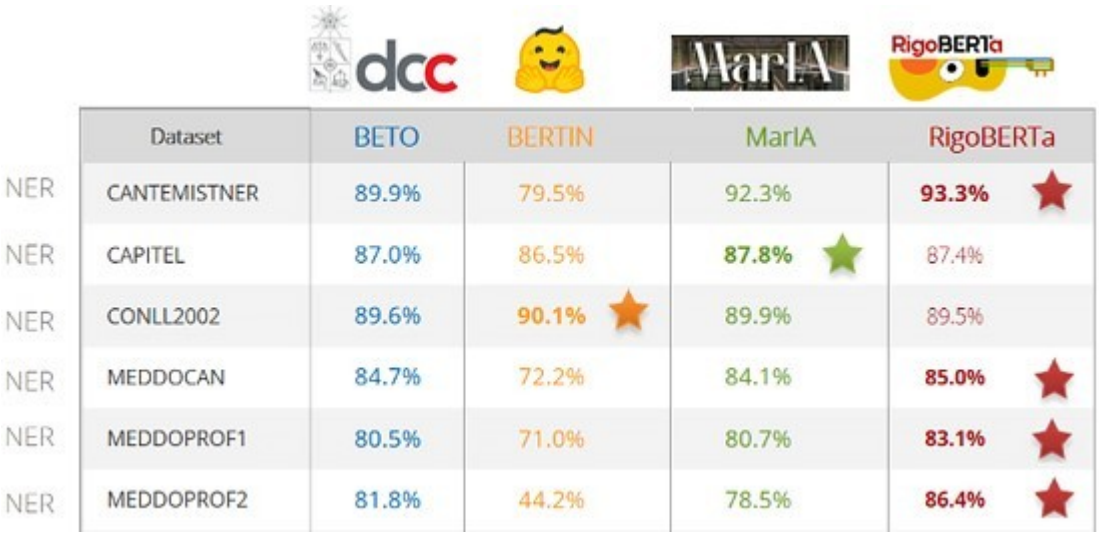
Desde la antigüedad hasta los tiempos actuales, la información ha experimentado un crecimiento exponencial en forma de documentos, libros y artículos, almacenándose en diferentes formatos: impresos, en forma electrónica (digital), con la llegada de las computadoras y el procesamiento del conocimiento el incremento ha ido en aumento. Sin embargo, lo que es conocimiento para nosotros los humanos no lo es para las computadoras. La computadora puede almacenar, respaldar, transmitir y borrar datos en forma de archivos, pero no puede buscar las respuestas a preguntas formuladas, hacer inferencias lógicas sobre su contenido, generalizar o resumirlo, porque no lo puede entender (Vásquez et al. 2009).

En el campo de las ciencias de la computación, la Inteligencia artificial está revolucionando nuestra forma de interactuar con lo que nos rodea, dicha tecnología se encuentra en métodos informáticos como el *Machine Learning*, en la interconexión de dispositivos conocido como *Internet of Things* y también en el Procesamiento del Lenguaje Natural (NLP) (Gamboa-Rosales et al. 2020). El NLP es un campo de las ciencias de la computación, inteligencia artificial y la lingüística que estudia las interacciones entre las computadoras y el lenguaje humano, por medio del análisis sintáctico, semántico, pragmático y morfológico; se escriben reglas de reconocimiento de patrones estructurales, empleando un formalismo gramatical concreto. Estas reglas, en combinación con la información almacenada en diccionarios computacionales, definen los patrones que hay que reconocer en una letra palabra u oración (Torres and Manjarrés-Betancur 2020).

Dentro del Procesamiento del Lenguaje Natural el Reconocimiento de Entidades Nombradas (NER) es la tarea de reconocer dentro del texto menciones de designadores rígidos pertenecientes a categorías semánticas predefinidas tales como ubicación, persona, organización, etc. (Nadeau and Sekine 2007) , también juega un rol importante en una gran variedad de aplicaciones del procesamiento del lenguaje natural tales como comprensión

textual, sumarización automática de textos, recuperación de información, respuestas automáticas, construcción de la base del conocimiento etc.

La prensa digital se caracteriza por mezclar en una misma plataforma diversas clases de formatos tales como texto, audio, gráficos, vídeos, animaciones y fotografías. Esta posibilidad ayuda a que las noticias se enriquezcan y se extiendan a nivel informativo, como resultado, la demanda de resultados de cierto tipo por parte de la industria, exigen procesos más económicos y eficientes de obtener los resultados, por lo que el empleo de técnicas de aprendizaje automático y de procesamiento de lenguaje natural han cobrado gran importancia en los últimos años(Falck et al. 2020) (Rusnachenko and Loukachevitch 2018) y herramientas como el Reconocimiento de Entidades Nombradas auxilian el procesamiento de esta gran cantidad de datos y posterior tratamiento(Figuerola et al.). Las herramientas actuales que usan Reconocimiento de Entidades Nombradas están enfocadas mayormente hacia un dominio especifico y los *datasets* más populares y comúnmente usados por estas herramientas se encuentran en idioma ingles (Albared et al. 2019), soluciones como los modelos multi lenguaje fueron creados y utilizados para combatir la barrera del lenguaje pero aunque tienen una aplicabilidad mayor generalmente su desempeño es peor(Toribio et al. 2010), la Figura 1 muestra los resultados de un *benchmark* comparativo entre varios modelos de lenguaje español disponibles públicamente.



	Dataset	BETO	BERTIN	MarIA	RigoBERTa
NER	CANTEMISTNER	89.9%	79.5%	92.3%	**93.3%** ★
NER	CAPITEL	87.0%	86.5%	**87.8%** ★	87.4%
NER	CONLL2002	89.6%	**90.1%** ★	89.9%	89.5%
NER	MEDDOCAN	84.7%	72.2%	84.1%	**85.0%** ★
NER	MEDDOPROF1	80.5%	71.0%	80.7%	**83.1%** ★
NER	MEDDOPROF2	81.8%	44.2%	78.5%	**86.4%** ★

Figura 1 Desempeño modelos en español

Fuente [Modelo de lenguaje en español - RigoBERTa (uam.es)]

A partir de la situación descrita anteriormente, se plantea el siguiente **problema a resolver**:

¿Cómo **extraer entidades nombradas de noticias en español de forma automática** de varios dominios?

.

El **objeto de estudio** donde se enmarca la investigación está constituido por la extracción de entidades nombradas dentro del **campo de acción** de textos en idioma español.

Para resolver el problema se identifica el siguiente **objetivo general**:

Desarrollar un componente de software que integre las herramientas y enfoques existentes para realizar la extracción de entidades nombradas dentro del campo de acción de textos en idioma español.

A partir de lo planteado anteriormente se desglosan los siguientes objetivos específicos:

1. Realizar el marco teórico y el estado del arte de la investigación para identificar tendencias y enfoques y adoptar una posición al respecto.

2. Realizar un análisis de los enfoques existentes y escoger el más adecuado para dar solución a la problemática

3. Implementar un componente de software para la extracción de entidades nombradas en español.

4. Validar que el componente de software implementado resuelve el problema de la extracción de entidades nombradas en textos en idioma español.

Se obtienen como **posibles resultados** un componente de software y su código fuente para la extracción de entidades nombradas a partir de noticias en español.

Se plantea como **idea a defender**:

Con la implementación de un componente de software que permita la extracción de entidade s nombradas de noticias en español se mejorarán los procesos de consulta, análisis y procesamiento de noticias de los medios de prensa nacionales y el Centro de Información para la Prensa.

Durante la investigación se han empleado un conjunto de métodos científicos como procedimientos lógicos, que se han seguido para la obtención y procesamiento de la información.

Métodos teóricos

El método **Analítico-Sintético** ha permitido realizar un análisis sobre la teoría y las tendencias de los componentes relacionados con el PLN y el NER, de manera que se hayan podido estudiar a profundidad cada uno de ellos por separado, así como las técnicas o tecnologías involucradas en el proceso de alineación que se realiza entre ambos. Ha permitido además caracterizar cada uno de los componentes analizados previamente.

El método **Histórico-Lógico** ha permitido analizar la evolución, de forma cronológica, de los elementos relacionados al NER, así como la evolución de tecnologías usadas.

Métodos empíricos

El método **observación** ha permitido obtener información relacionada con el comportamiento de las herramientas y las tecnologías existentes para la extracción de entidades nombradas de noticias permitiendo que se use esa información para el desarrollo de un método que permita la extracción de entidades nombradas de noticias en español.

El método **experimentación** ha permitido la realización de experimentos para validar la propuesta de solución.

El método de la **medición** ha permitido la evaluación de los resultados de la aplicación de pruebas a la solución.

La investigación está estructurada en tres capítulos

Capítulo I: Se definen los principales conceptos, relacionados en el trabajo, que pertenecen al ámbito del Procesamiento del Lenguaje Natural y la Extracción de Entidades Nombradas. Se realiza un estudio de la literatura para identificar los elementos que formarán parte de la propuesta de solución.

Capítulo II: Se define el método para la extracción de entidades nombradas en textos en idioma español. Se exponen las características relevantes en cuanto a su implementación.

Capítulo III: Se describe el proceso de validación de la propuesta de solución. Son realizadas pruebas para validar el diseño y la herramienta. Se expone un caso de estudio para validar el método para la extracción de entidades nombradas en textos en idioma español., haciendo un análisis de las variables tiempo y recursos computacionales. Se ilustran ejemplos y resultados finales.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO

I.1 Introducción.

En este capítulo se hace referencia a los fundamentos básicos de la Extracción de Entidades Nombradas (NER) y a los elementos necesarios para la alineación entre NLP y NER, haciendo énfasis en los conceptos, definiciones y características principales. Se realiza un análisis de la literatura para identificar los elementos que formarán parte de la propuesta de solución. Se establece una comparación entre los enfoques y herramientas utilizadas en los procesos de extracción de entidades nombradas. Se identifican la metodología, las herramientas, tecnologías y otros elementos utilizados en la confección de la solución.

I.2 Marco teórico. Conceptos y definiciones

I.2.1 Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural consiste en la habilidad de una máquina para procesar información comunicada mediante el uso del lenguaje natural. Crean modelos computacionales del lenguaje suficientemente detallados que permitan escribir programas informativos que realicen distintas órdenes o peticiones donde interviene el lenguaje natural. Se podría decir que el NLP consiste en usar una expresión natural que pueda tener comunicación con la computadora directamente, por medio escrito o comando de voz, facilitando las órdenes o peticiones con el lenguaje, o seguir desarrollando modelos que ayuden a la comprensión humana y sus mecanismos que se relacionan al lenguaje(Gelbukh 2010; Moreira et al. 2021).

Dentro del campo del procesamiento del lenguaje natural se pueden diferenciar dos aproximaciones principales: procesamiento estadístico del lenguaje natural, y el procesamiento lingüístico del lenguaje natural, en inglés conocido como *rule-based Natural Language Processing(Nivre 2001)*.

I.2.2 Procesamiento estadístico del Lenguaje Natural

El enfoque estadístico se caracteriza por la elaboración de los sistemas que no son utilizados para el almacenamiento del conocimiento lingüístico, o del mundo; empleando técnicas de tratamiento de información con la finalidad de sacar el potencial del conocimiento del mundo. Los procesos del lenguaje están basados en modelos formales del conocimiento lingüístico, entre los relevantes están: máquinas de estado, sistemas de reglas, lógica, o modelos probabilísticos. Actualmente se ha demostrado que no todos los sistemas NLP toleran adaptarse a teorías lingüísticas, ni si quiera los de procesamiento simbólico(Moreira, Cruz, Gonzalez, Quirumbay, Magallan, Guarda, Andrade and Castillo 2021; Pascual 2012).

I.2.3 Procesamiento lingüístico del Lenguaje Natural

El enfoque simbólico se caracteriza por la elaboración de sistemas que son utilizados para almacenar específicamente los actos lingüísticos, por ejemplo: fonológicos/fonéticos, morfológicos, sintácticos, semánticos, pragmáticos o discursivos, que, mediante esquemas de representación del conocimiento, elaborados con la finalidad de manual(Pascual 2012)

I.1.1¿Qué es NER?

Una entidad nombrada es una palabra o una frase que identifica claramente un elemento de un conjunto de otros elementos que tienen similares atributos(Sharnagat 2014). Ejemplos de entidades nombradas son organización, nombres de personas y lugares en el dominio general; nombres de genes, proteínas, fármacos y enfermedades en el dominio biomédico. NER es el proceso de localizar y clasificar entidades nombradas en el texto en categorías de entidades predefinidas.

El término “Entidad Nombrada” (NE) fue utilizado por primera vez en la sexta Conferencia de Comprensión de Mensajes(MUC-6)(Grishman and Sundheim 1996), como la tarea de identificar nombres de organizaciones, personas y ubicaciones geográficas en el texto, así como como expresiones de moneda, tiempo y porcentaje. Desde MUC 6 ha habido un creciente interés en NER, y varios eventos científicos (p. ej., CoNLL03(Akbik et al. 2019),

ACE(Wang et al. 2020), IREX(Wang and Su 2022), y TREC *Entity Track(Missaoui et al. 2019)*) dedican mucho esfuerzo a este tema. En cuanto a la definición del problema, (Petasis et al. 2000) restringió la definición de entidades nombradas: “UN NE es un nombre propio, que sirve como nombre para algo o alguien”. Esta restricción se justifica por el importante porcentaje de nombres propios presentes en un corpus. (Nadeau and Sekine 2007) afirmó que la palabra "Nombrado" restringió la tarea a aquellas entidades para las cuales uno o muchos designadores rígidos representan para el referente. El designador rígido, definido en(Kripke 1972), incluye nombres propios y términos de género natural como especies biológicas y sustancias. A pesar de las diversas definiciones de NE, Los investigadores han llegado a un consenso común sobre los tipos de NEs a reconocer. Generalmente dividimos los NE en dos categorías: NE genéricos (por ejemplo, persona y ubicación) y NE específicos de dominio (p. ej., proteínas, enzimas y genes).

Formalmente, dada una secuencia de tokens s = ⟨w1, w2, ..., wN ⟩, NER genera una lista de tuplas ⟨Is, Ie, t⟩, cada una de las cuales es una entidad nombrada mencionada en s. Aquí, Is ∈ [1, N] e Ie ∈ [1, N] son los índices inicial y final de una mención de entidad nombrada; t es el tipo de entidad de un conjunto de categorías predefinidas. La Figura 1 muestra un ejemplo donde un sistema NER reconoce tres entidades nombradas de la sentencia dada.

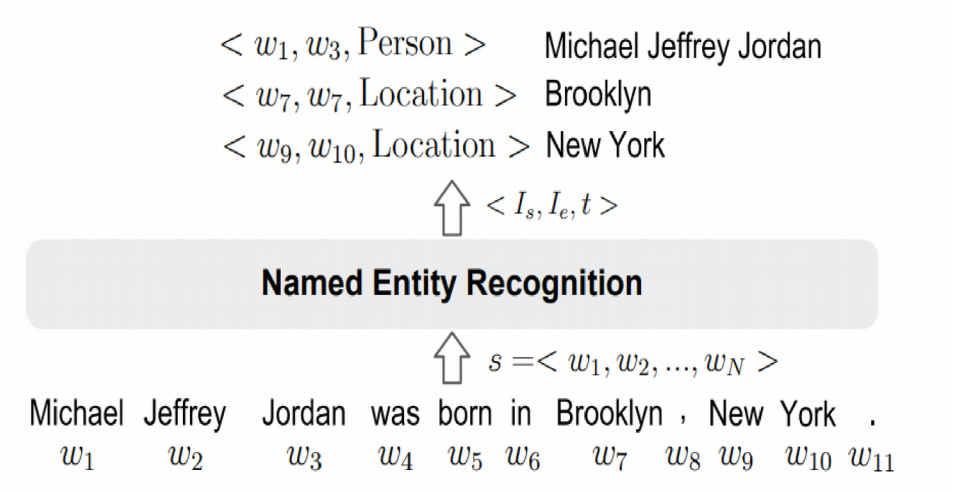


Figura 2 Una ilustración de la tarea de reconocimiento de entidad nombrada

NER actúa como un importante paso de preprocesamiento para una variedad de aplicaciones posteriores, como recuperación de información, respuesta a preguntas, traducción automática, etc. En la Figura 1, se usó la búsqueda semántica como ejemplo para ilustrar la importancia de NER en el soporte de diversas aplicaciones. La búsqueda semántica se refiere a una colección de técnicas, que permitir que los motores de búsqueda entiendan los conceptos, el significado, y la intención detrás de las consultas de los usuarios(Balog 2018).

I.2.1 Métricas de evaluación NER

Los sistemas NER generalmente se evalúan comparando sus Salidas contra anotaciones humanas. La comparación puede ser cuantificado por coincidencia exacta o coincidencia relajada.

I.2.I Evaluación de coincidencia exacta

La evaluación de coincidencia exacta implica identificar tanto los límites de la entidad como el tipo de entidad. Con la evaluación de coincidencia exacta, una entidad nombrada se considera correctamente reconocido sólo si sus dos límites y tipo se encuentran en el terreno válido(Sang and De Meulder 2003), Precisión, Reconocimiento, y la puntuación F se calculan sobre el número de verdaderos positivos (VP), falsos positivos (FP) y falsos negativos (FN).

- Verdaderos positivos (VP): entidades que son reconocidas por NER y se encuentran dentro del terreno válido
- Falso positivo (FP): entidades que son reconocidas por NER pero que no se encuentran dentro del terreno válido.
- Falso Negativo (FN): entidades dentro del terreno válido que no son reconocidos por NER.

La Precisión mide la habilidad de un sistema NER para presentar solo entidades correctas, y el Reconocimiento mide la habilidad de reconocer todas las entidades en un corpus.

Precisión = Verdaderos positivos /Verdaderos positivos + Falso positivo

Reconocimiento = Verdaderos positivos /Verdaderos positivos +Falso Negativo

La puntuación F es la media armónica entre Precisión y Reconocimiento y la puntuación F balanceada es la más comúnmente usada:

Puntuación F = 2 × Precisión × *Recall*/Precisión + *Recall*

I.2.2 Evaluación de coincidencia relajada

MUC-6(Grishman and Sundheim 1996) define una evaluación de partido relajado: se acredita de tipo correcto si a una entidad se le asigna su tipo correcto independientemente de sus límites, siempre que haya una superposición con límites los del terreno válido ; un límite correcto es acreditado independientemente de la asignación de tipo de una entidad. Entonces ACE(Doddington et al. 2004) propone un procedimiento de evaluación más complejo. Resuelve unos pocos problemas como coincidencia parcial y tipo incorrecto, y considera subtipos de entidades nombradas. Sin embargo, es problemático porque los puntajes finales son comparables solo cuando los parámetros son fijos(Goyal et al. 2018; Sun et al. 2018). Los métodos de evaluación complejos no son intuitivos y dificultan el análisis de errores.

I.3 Estado del Arte

Las técnicas aplicadas en el reconocimiento de entidades han evolucionado desde la elaboración manual de patrones, donde los recursos son etiquetados manualmente, hasta reglas obtenidas automáticamente mediante técnicas de aprendizaje.

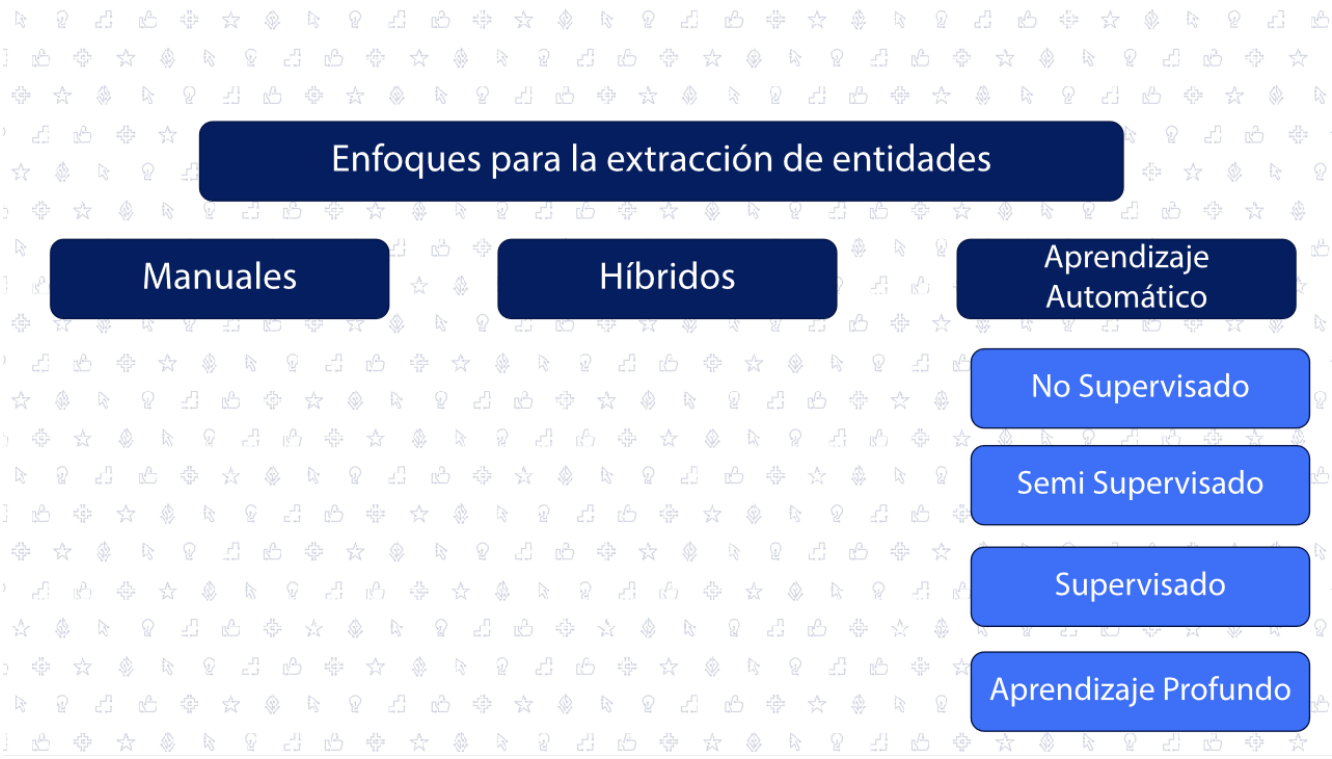


Figura 3 Clasificación de Sistemas de Extracción de Entidades

I.3.1 Sistemas basados en reglas (HANDCRAFTED).

Estos sistemas son construidos a mano. Ayudados por un conjunto de reglas y heurísticas que guían el proceso de etiquetado, están basados en el conocimiento de los especialistas que los diseñan. Los recursos más utilizados para estos sistemas son las expresiones regulares, conjuntos de reglas y *Gazetteers* (listas de palabras)(Rijhwani et al. 2020). Generalmente están compuestos por conjuntos de patrones con características gramaticales, sintácticas y ortográficas en combinación con diccionarios(Liu et al. 2019). Los sistemas manuales tienen la ventaja de poder obtener muy buenos resultados. Una gran mayoría de ellos alcanza una precisión del 90%, aunque para mayor rendimiento sea necesario definir manualmente un conjunto de reglas por cada par lenguaje/dominio. Tanto es así que, estos tipos de modelos tienen mejores resultados para dominios restringidos que los modelos basados en aprendizaje automático. Son capaces de detectar entidades complejas cosa que presenta un problema para los sistemas automáticos. Sin embargo, este tipo de enfoques carece de la habilidad de hacer frente a los problemas de robustez y portabilidad, tan

necesarios a día de hoy. Cada texto o fuente nueva requiere una revisión de las reglas para mantener un rendimiento óptimo. Lo que quiere decir que cada vez que haya un nuevo dominio y/o lenguaje, habría que reescribir el sistema. Además, hay que añadir que el coste de mantenimiento podría ser muy elevado.

I.3.2 Sistemas híbridos.

La estrategia detrás de los sistemas híbridos consiste en combinar las fortalezas de los sistemas manuales con aquellos basados en aprendizaje automático para reducir sus debilidades. La mayoría de los sistemas analizados son, hasta cierto punto, híbridos pues, aunque se basen en modelos de aprendizaje automático, hacen uso de recursos lingüísticos como *Gazetteers* o listas de palabras. A continuación, se describen algunos ejemplos de sistemas híbridos:

- LTG(Mansouri et al. 2008), hace uso de evidencia interna y externa basándose en la idea de que ciertas cadenas tienen una estructura que sugiere que son ENs pero no de qué tipo. La idea principal es retrasar la clasificación final de un NE hasta que esa pieza de información contextual sea encontrada y las ENs sean desambiguadas. ⟩
- MENE(Chiong and Wei 2006), es otro sistema con buen rendimiento. Utiliza un clasificador de entropía máxima para combinar salidas de varios sistemas hechos a mano y obtiene resultados superiores a los obtenidos por cada uno de dichos sistemas independientes. El etiquetado de ENs se aborda como un etiquetado de secuencias donde varias características (internas, externas, locales, globales) son desarrolladas y combinadas ⟩

ME & HMM, presentado por (Srihari 2000), combinan ME (entropía máxima), HMM (*Hiden Markov Model*) y reglas gramaticales hechas a mano. Aunque cada método tiene sus debilidades, la combinación de ellos dio como resultado un etiquetador de alta precisión, además incluyen *Gazetteers* internos. ⟩

I.3.3 Sistemas basados en aprendizaje automático.

Tiene como objetivo desarrollar técnicas que sean capaces de generalizar comportamientos a partir de una información estructurada que es suministrada en forma de ejemplos. Las

investigaciones sobre algoritmos de aprendizaje automáticos fueron motivadas a partir de la posibilidad de no necesitar gente especializada en el campo para examinar los datos y encontrar patrones o reglas asociados a éstos. De hecho, actualmente, la tendencia de la EEN es utilizar métodos de aprendizaje automático, puesto que se adaptan con mayor facilidad a distintos dominios.

I.3.2.1 Métodos de aprendizaje supervisados.

Para utilizar algoritmos de aprendizaje supervisado es necesario contar con un corpus donde las entidades estén anotadas para que sirvan como ejemplo en el entrenamiento. La tarea de etiquetar el corpus requiere un esfuerzo considerable, no siendo necesario un grado de especialización alto. Sin embargo, para el personal que define los conjuntos de las reglas en los sistemas manuales ha de ser considerable. Dentro de los sistemas que utilizan aprendizaje automático, encontramos los basados en reglas y los basados en métodos estadísticos. Los métodos basados en reglas son de fácil comprensión, desarrollo y expansión, además de rápidos y útiles para tareas controladas, como la extracción de números de teléfono, códigos postales de correos electrónicos, etc. Están formados por dos partes, una colección de reglas y las políticas necesarias para controlar el disparo de esas reglas. Las reglas pueden ser aprendidas mediante técnicas de aprendizaje automático, a partir de ejemplos etiquetados en textos no estructurados o bien con métodos *Handcrafted*. El objetivo de este método es conseguir un conjunto de reglas tales que posean una buena precisión y cobertura en nuevos documentos. Existen dos modos:

-*Bottom-Up*: como el algoritmo (LP)(Sintayehu and Lehal 2021), parten de una regla muy específica con el 100% de precisión y cobertura mínima consiguiendo generalizarse a través del aprendizaje. }

-*Top-Down*. Basado en algoritmos como FOIL(Khadir et al. 2021) y WHISK(Nasar et al. 2021) parten de una regla muy genérica y se especializan con el aprendizaje.

Los métodos basados en estadísticas (Gaytán Díaz 2018) convierten la tarea de extracción en un problema de descomposición del texto no estructurado etiquetando las partes de la descomposición bien de forma conjunta o independiente. Según este tipo de descomposición se puede distinguir tres tipos de métodos: *Token-level(Hollenstein and Zhang 2019)*,

Segmentlevel y métodos basados en gramáticas. Todos ellos reciben como entrada parte de un texto no estructurado compuesto por Tokens y de un conjunto de tipos de entidades que se quieren extraer de él.

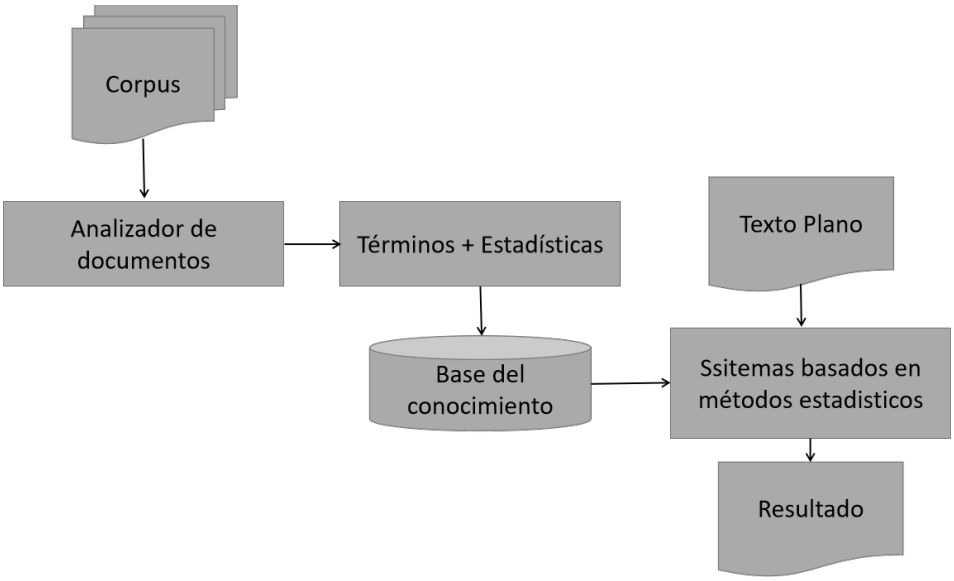


Figura 4 Métodos basados en estadísticas.

En los modelos *Token-level* el texto no estructurado se trata como una secuencia de tokens, reduciendo la extracción a asignar a cada token una etiqueta de entidad. Cuando todos los tokens están etiquetados, las entidades se marcan como tokens consecutivos de la misma etiqueta entidad. Las propiedades que se utilizan en estos modelos suelen ser propiedades de palabra, ortográficas y de aparición en diccionarios. Para la clasificación la pareja token-entidad se suelen utilizar los modelos ocultos de Harkov *(HMM-Hiden Markov Model*-primer sistema NER estadístico de alto rendimiento), etiquetadores de entropía máxima (MEMM-*Maximum Entropy Markov*), modelos condicionales de Markov (*CMM-Conditional Markov Models*) y los CRF (*Conditional random fields*) que presentan mejores resultados que los métodos lineales (como SVM-*Suppor Vector Machine*) tanto teórica como empíricamente

En los modelos *segment-level*, la salida es una secuencia de segmentos, *chunks* donde cada uno define una entidad. Las características se definen sobre segmentos compuestos por

múltiples Tokens que forman una cadena completa de entidad. Esto le permite utilizar atributos más potentes que los *token-level*. Así, además de utilizar los *token-level*, se pueden usar parámetros con similitud a una entidad existente en una base de datos y la longitud del segmento de la entidad. Estos modelos son efectivos para frases bien formadas o estructuras del lenguaje natural. Si tratan de explotar la estructura global del texto fuente fallan, al no seguir el formato de sentencia bien formada. Esto se puede remediar con el uso de modelos basados en gramáticas libres dado que son más efectivos para estructuras fuera que se encuentren al margen de frases bien formadas. Ocurre porque usan un conjunto de reglas de producción para expresar la estructura global de la entidad dando como salida un árbol de parseado o análisis. A pesar de que estos modelos basados en segmentos y en gramáticas proporcionan toda la flexibilidad de los CRF's, no son muy populares debido al aumento de coste en la inferencia y entrenamiento. Tampoco los métodos puramente automáticos basados en datos para la inducción de reglas lo son debido a la limitada existencia de datos etiquetados. Es por ello por lo que los sistemas basados en reglas más exitosos hacen uso de métodos híbridos, donde se extraen reglas semilla de los datos etiquetados para después modificar o afinar esas reglas.

I.3.2.2 Sistemas basados en aprendizaje semi-supervisado.

El etiquetado manual de un corpus es tarea costosa, por lo que no siempre se puede contar con la disponibilidad de suficientes muestras de entrenamiento para los clasificadores. Sin embargo, existen cantidad de textos sin etiquetar a los que se pueden acceder de forma fácil y económica. El objetivo del aprendizaje semi-supervisado es combinar muestras etiquetadas y no etiquetadas para mejorar los clasificadores. La técnica más popular es el ‘’*bootstrapping*’’ (Chaudhary et al. 2019), el cual requiere un conjunto de semillas que usa al principio para buscar oraciones o instancias que coincidan con estas semillas.

Otro método semi- automático es el llamado Aprendizaje Activo (*Active Learning* -AL), donde es el propio sistema el que proporciona al usuario los candidatos para que los corrija o etiquete, de modo que puedan ser utilizados como nuevas semillas para reentrenar el modelo(Liu et al. 2020). Existen diversas técnicas para la selección de los ejemplos a etiquetar(Mi et al. 2020):

Uncertainty sampling: se seleccionan las instancias con menos certeza de que sean válidas.

Query by Committee: cuando el aprendizaje se realiza con varios algoritmos a la vez, se seleccionan las instancias donde más desacuerdo exista entre ellos. ⟩

Density-Weighted Methods: se seleccionan las instancias no sólo con menos certeza de validez, sino que además son representativas de la distribución de entrada. De este modo se trata de evitar la selección de outliners a la que son propensos los métodos anteriores. ⟩

Expected Model Change: se seleccionan aquellas instancias que más puedan influir en el modelo. ⟩

Variance Reduction and Fisher Information Ratio: se eligen las instancias bajo el criterio general de minimizar la varianza. ⟩

Estimated Error Reduction: se seleccionan las instancias que minimizan el error esperado.

I.3.2.3 Métodos de aprendizaje no supervisado.

Este tipo de métodos no necesitan ejemplos de entrenamiento, no son muy populares para la extracción de entidades debido a que los sistemas que lo usan no han alcanzado el nivel de desempeño de los supervisados. Por ello, los sistemas que aplican este enfoque usualmente no son completamente no-supervisados, sino que tienden a ser híbridos. Generan un conjunto de reglas a partir de la combinación de módulos de aprendizaje supervisado y no supervisados con el objetivo es construir representaciones de los datos. Lo métodos no supervisados pueden ser fácilmente portados a diferentes dominios y lenguajes. El enfoque típico para este tipo de aprendizaje es el *clustering*, dado que trata de extraer ENs de *clusters* basados en la similitud del contexto. Básicamente las técnicas recaen en recursos y patrones léxicos y en estadísticas calculadas en corpus grandes no etiquetados(Shaffer 2021).

I.3.4 Aprendizaje Profundo.

Existen diversas definiciones para el concepto de Aprendizaje Profundo (*Deep Learning*), pero la mayormente se trata de un campo de *Machine Learning* basado en algoritmos capaces de modelar problemas en abstracciones de múltiples capas y generalmente asociado al uso de redes neuronales multicapa o redes neuronales profundas(Deng et al.

2014). Algunas de las soluciones actuales a la tarea NER se han realizado con el uso de *Deep Learning*, específicamente con el uso de redes neuronales recurrentes (Li et al. 2020; Yadav and Bethard 2019). A continuación, se presentarán algunos conceptos básicos asociados a la definición, diseño y entrenamiento de este tipo de modelos.

I.3.4.I Perceptrón.

El perceptrón es la unidad básica de una red neuronal. Esta estructura fue definida en la década de los 50 (Deng, Yu and processing 2014) y posee cierta inspiración en la biología, en el funcionamiento de la neurona, la cual es capaz de recibir múltiples entradas (*inputs*), cuya sumatoria es recibida por una función de activación que definirá la salida de la unidad. Una red neuronal artificial está formada por múltiples perceptrones interconectados(Haykin 2009). En la Fig.4, se definen diferentes elementos para un perceptrón.

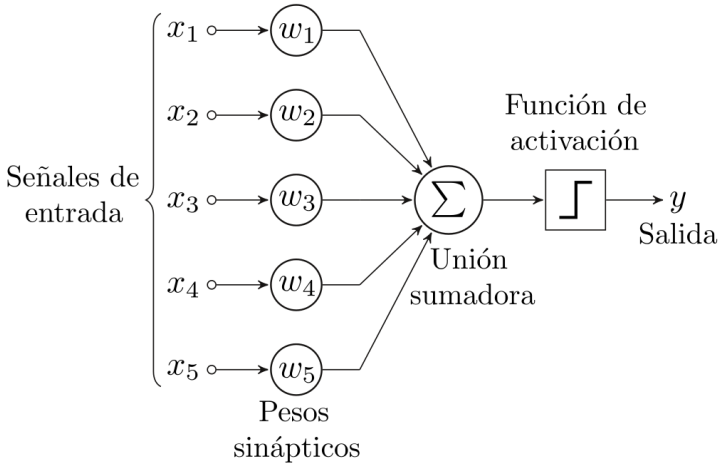


Figura 5 Modelo de Perceptrón

Fuente Haykin S. (2009) “Neural Network and Learning Machines”

I.3.4.2 Red Neuronal Multicapa.

Una red neuronal está formada por capas de uno o más perceptrones y a su vez, de múltiples capas. La cantidad de capas que se asocia a la profundidad de la red(Haykin 2009).

La red neuronal *feedforward* de tres niveles de la Fig. 5, cuenta con una capa de entrada de diez nodos, una intermedia (u oculta) de cuatro neuronas y una capa de salida (*output*) de dos neuronas (Haykin 2009).

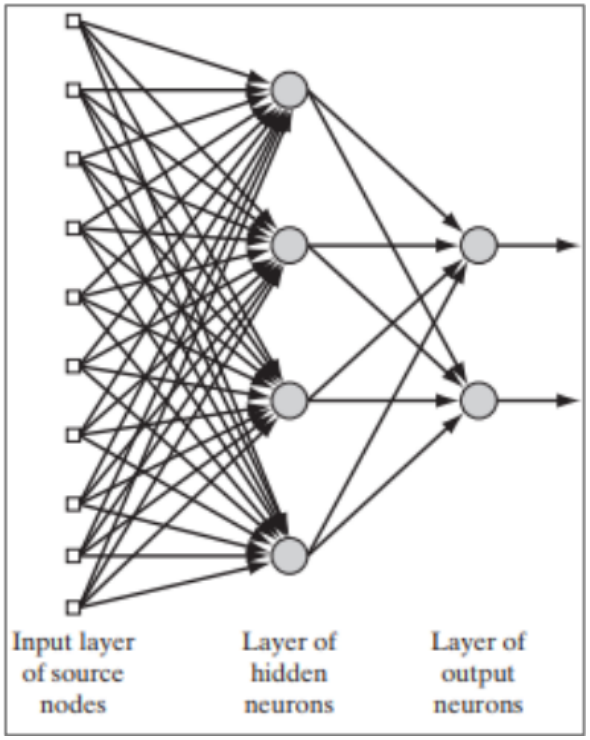


Figura 6 Feedforward fully-connected

Fuente Haykin S. (2009) “Neural Network and Learning Machines”

I.3.4.3 Redes Neuronales Recurrentes.

Las redes neuronales recurrentes o RNN (*Recurrent Neural Network*), fueron propuestas por primera vez a principio de los años 90’s(Elman 1990). La RNN simple propuesta por Elman en ese entonces, consideraba una copia directa de la activación de la capa oculta en unidades concatenadas a la capa de entrada y denominadas “unidades de contexto”. Este tipo de redes están formadas por un nuevo tipo de perceptrón que acepta dos tipos de entrada; la actual y la salida previa de la unidad. De esta forma una neurona recurrente transmite la información hacia adelante pero también tiene la característica de enviar la información hacia atrás. Por lo tanto, en cada paso, la neurona recurrente recibe datos de las neuronas anteriores, pero también recibe información de ella misma en el paso anterior.

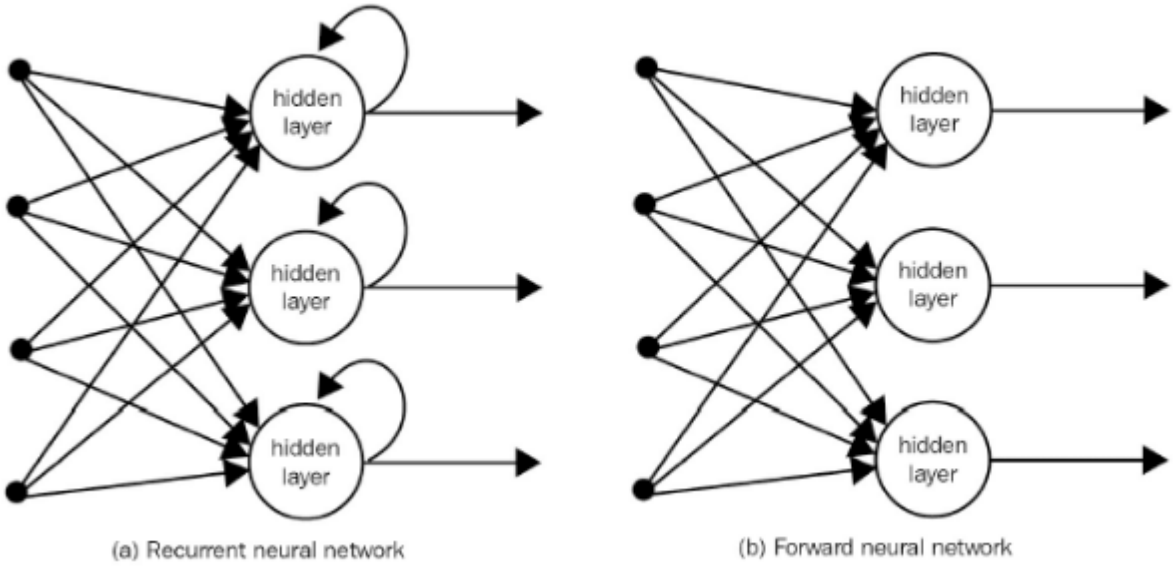


Figura 7 Red neuronal recurrente.

PREDICCION DE LA CALIDAD DE SOFTWARE DESARROLLADO EN IBM RPG USANDO DEEP LEARNING - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Figura-24-Red-neuronal-recurrente_fig8_328600321 [accessed 24 Oct, 2022]

I.4 Herramientas para la extracción de entidades usando *Deep learning*.

1.4.1 *PyTorch*

PyTorch es un *framework* de aprendizaje automático de código abierto relativamente nuevo basado en *Torch*, utilizado para aplicaciones que implementan cosas como visión artificial y procesamiento de lenguajes naturales y desarrollado por el grupo de investigación de *Facebook*(FAIR). *PyTorch* tiene una reputación de simplicidad, facilidad de uso, flexibilidad, uso eficiente de la memoria y gráficos computacionales dinámicos. También se siente nativo, lo que hace que la codificación sea más manejable y aumenta la velocidad de procesamiento.

Un gran número de las piezas de software de Aprendizaje Profundo están construidas utilizando *PyTorch*, incluyendo *Tesla Autopilot*, *Uber's Pyro*, *HuggingFace's Transformers*, *PyTorch Lighting*, y *Catalyst(Ketkar and Moolayil 2021; Vasilev et al. 2019)*.

1.4.2 *Keras*

Keras es una interfaz de programación de aplicaciones (API) de red neuronal efectiva de alto nivel escrita en *Python*. Esta biblioteca de redes neuronales de código abierto está diseñada para proporcionar una experimentación rápida con redes neuronales profundas y puede ejecutarse sobre CNTK, *TensorFlow* y *Theano*.

Keras se enfoca en ser modular, fácil de usar y extensible. No maneja cálculos de bajo nivel; en cambio, los entrega a otra biblioteca llamada *Backend(Gulli and Pal 2017; Manaswi 2018; Vasilev, Slater, Spacagna, Roelants and Zocca 2019)*.

1.4.3 *Tensorflow*

Tensorflow es un *framework* de aprendizaje profundo de código abierto de extremo a extremo desarrollado por *Google* y lanzado en 2015. Es conocido por su soporte de documentación y capacitación, opciones de producción e implementación escalables, múltiples niveles de abstracción y soporte para diferentes plataformas, como Android.

Tensorflow es una biblioteca matemática simbólica utilizada para redes neuronales y es más adecuada para la programación de flujo de datos en una variedad de tareas. Ofrece múltiples niveles de abstracción para construir y entrenar modelos(Vasilev, Slater, Spacagna, Roelants and Zocca 2019).

1.4.4 Comparación de herramientas para procesamiento del lenguaje natural con aprendizaje profundo.

	PyTorch	***Keras***	***TensorFlow***
Nivel de la API	Bajo	Alto.	Alto y bajo.
Arquitectura	Complicada, menos compresible	Simple, concisa, legible.	No es fácil de usar.
Datasets	*Datasets* extensos, alto rendimiento.	*Datasets* más pequeños.	*Datasets* extensos, alto rendimiento.
Depuración	Capacidad de depuración apreciable	La red es simple, por lo que la depuración	Es difícil ejecutar la depuración.

		no es necesaria frecuentemente.	
Posee modelos entrenados	Sí.	Sí.	Sí.
Popularidad	Popularidad baja	Muy popular.	Popularidad media
Velocidad	Alto rendimiento	Bajo rendimiento.	Alto rendimiento.
Escrito en	*C++, Python, CUDA, C*	*Python,CUDA*	*Python, C++,CUDA*
Código abierto	Sí	Si	Sí
Licencia	BSD	Licencia MIT	Apache 2.0
Creado por	*Meta AI*	François Chollet, *Google*	Equipo *Google Brain*

Tabla 1 Comparación herramientas deep learning

Las particularidades de la problemática requieren de herramientas de alto rendimiento y versatilidad, *TensorFlow fue escogida como herramienta a utilizar por su velocidad y consistencia.*

Conclusiones del capítulo

El área del Extracción de la Información cuenta actualmente con diversas aplicaciones en ámbito comercial y científico principalmente. Sus sistemas se desarrollan con combinación de técnicas de procesamiento de lenguaje natural y aprendizaje automático para la realización de sus principales subtareas.

La tarea de la extracción de entidades nombradas es abordada por diversos enfoques: de forma manual, automática o híbrida, estos difieren en las técnicas usadas, tiempo de resolución y calidad en los resultados obtenidos. Se arriba a la conclusión que los algoritmos de aprendizaje profundo dentro del aprendizaje automático se adhieren a las particularidades del problema expuesto y arrojan los mejores resultados luego de consultada la literatura.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

2.1 Introducción

El este capítulo se describe la propuesta de solución desde un enfoque teórico, basada en una metodología para proyectos de minería de datos para la extracción de entidades nombradas de noticias en español. En la segunda sección se describe la arquitectura, componentes, tecnologías y algoritmos aplicados en el desarrollo de un prototipo funcional que implementa el método propuesto.

2.1 Metodología.

Las técnicas de *Data Science* o *Data Analytics* surgieron en la década de los 90 en un intento de normalizar el proceso de descubrimiento de conocimiento. *Frameworks* como el Proceso Unificado (UP) o Scrum son estándares hoy en día en cualquier proyecto software y, en el caso de los proyectos *Big Data*, tenemos metodologías KDD (*Knowledge Discovery in Databases)* como CRISP-DM (*Cross Industry Standard Process for Data Mining*) y SEMMA (*Sample, Explore, Modify, Model, and Assess*) que nos ayudan a encontrar conocimiento en nuestros datos.

(Azevedo and Santos 2008) compara estas metodologías y llega a la conclusión de que, aunque se puede establecer un paralelismo claro entre ellas, *CRISP-DM* es más completo porque tiene en cuenta la aplicación al entorno de negocio de los resultados.

CRISP-DM (*Cross Industry Standard Process for Data Mining*) proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos, de forma análoga a como se hace en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software. El modelo *CRISP-DM* cubre las fases de un proyecto, sus tareas respectivas, y las relaciones entre estas tareas.

De manera parecida a UP y Scrum, CRISP-DM define un ciclo de vida enfocado a la **exploración y análisis de los datos**. Este ciclo de vida consta de 6 fases: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación y Despliegue. A continuación, os describimos cada una de las fases:

1. Compresión del negocio

Esta fase inicial se enfoca en la comprensión de los objetivos y exigencias del proyecto desde una perspectiva de negocio. Posteriormente convierte ese conocimiento de los datos en la definición de un problema de minería de datos y en un plan preliminar diseñado para alcanzar los objetivos.

2. Comprensión de los datos

La comprensión de los datos se encarga de la recolección de datos inicial y continúa con las actividades que permiten familiarizarse primero con los datos, identificar sus problemas de calidad, descubrir conocimiento preliminar en los mismos, y/o descubrir subconjuntos interesantes para formular hipótesis. En esta fase se tienen en cuenta también las fuentes de datos que hasta el momento no se estaban utilizando como fuentes externas.

3. Preparación de los datos

La fase de preparación de los datos cubre todas las actividades necesarias para construir el conjunto de datos final, los datos que serán provistos por las herramientas de modelado. Las tareas de preparación incluyen la selección de los datos, la limpieza de éstos, la construcción de nuevas variables, la integración de los datos y el formateo de estos.

4. Modelado

Durante esta fase, se aplican las técnicas de minería de datos a nuestros datos. Se aplican varias técnicas de modelado y los parámetros de uso de estas se afinan hasta alcanzar los valores óptimos. Algunas técnicas de modelado necesitan requerimientos específicos sobre el formato de los datos, que podrán llevarnos de nuevo a la fase de preparación de los datos.

5. Evaluación

En este caso se evalúan los modelos anteriores para determinar si son útiles a las necesidades de negocio. En esta etapa los modelos ya están construidos y deben tener una alta calidad desde una perspectiva de análisis de datos.

6. Despliegue

La fase de despliegue implica la explotación de los modelos dentro de un entorno de producción. La creación de un modelo no es generalmente el final del proyecto, podría ser necesario rehacer el modelo para tener en cuenta nuevo conocimiento en el futuro.

El uso de metodologías como CRISP-DM en proyectos Big Data no sólo agilizará su desarrollo, sino que, además, nos asegura calidad en los datos con los que trabajamos y los resultados que obtengamos.

2.2.1 Método propuesto.

Los métodos son los pasos que seguir para hacer algo. Con los métodos se ponen en práctica las teorías. En otras palabras, los métodos son caminos para llegar a un fin; implican actuar de una forma ordenada y calculada. El método sigue un conjunto de reglas que dan un orden.

El primer paso de una tarea NER es detectar una entidad. Esto puede ser una palabra o un grupo de palabras que se refieren a la misma categoría.

Para asegurarnos de que el modelo detecte que una entidad puede ser una sola palabra o un grupo de palabras, entonces necesitamos proporcionar información sobre el comienzo y el final de una entidad en nuestros datos de entrenamiento a través del llamado etiquetado *Inside-Outside-Beginning* (IOB)(Chhabra et al. 2022).

Después de detectar una entidad, el siguiente paso en una tarea NER es categorizar la entidad detectada. Las categorías de una entidad pueden ser cualquier cosa dependiendo de nuestro caso de uso. A continuación, se muestra un ejemplo de categorías de entidades:

Persona: Pérez, Jose Pérez, Samuel, Ana, Frank, Leonardo DiCaprio

Localización: Habana, Viena, México, Londres

Organización: Etecsa, *Apple*, Insituto de Meteorología

Ubicación: Parque Central, Plaza de la Revolución, Avenida Principal

2.3 Pre procesamiento

El propósito del pre procesamiento de datos es principalmente corregir las inconsistencias de los datos que serán la base de análisis en procesos de minería de datos. En el caso de las fuentes de datos estructuradas, el propósito no es distinto y pueden ser aplicadas diversas técnicas estadísticas y de aprendizaje computacional(Hernández and Rodríguez 2008).

Con el pre procesamiento de datos se pretende que los datos que van a ser utilizados en tareas de análisis o descubrimiento de conocimiento conserven su coherencia.

Antes de que podamos usar un modelo para clasificar la entidad de un token de manera concisa, primero debemos hacer el pre procesamiento de los datos, que incluye dos partes: tokenización y ajuste de la etiqueta para que coincida con la tokenización.

2.3.1 Tokenización.

En NLP el proceso de convertir nuestras secuencias de caracteres, palabras o párrafos en inputs para la computadora se llama tokenización. Se puede pensar al token como la unidad para procesamiento semántico.

El primer paso es dividir el texto en palabras (o partes de palabras, símbolos de puntuación, etc.), llamadas tokens. Hay múltiples reglas que pueden gobernar ese proceso, por lo que necesitamos instanciar el tokenizador usando el nombre del modelo, para asegurarnos de que usamos las mismas reglas que se usaron cuando se preentrenó el modelo(Rai and Borah 2021).

El segundo paso es convertir esos tokens en números, para poder construir un tensor con ellos y alimentar el modelo. Estos resultados, una vez convertidos en el tensor del marco apropiado, pueden utilizarse como entradas de un modelo(Rai and Borah 2021; Song et al. 2020).

2.3.2 Ajuste de la etiqueta después de la tokenización.

Existen dos problemas que debemos abordar después del proceso de tokenización:

1. La adición de tokens especiales de BERT.

2. El hecho de que algunos tokens se dividen en subpalabras.

Al usar tokenización de subpalabras, la tokenización de piezas de palabras(*word-piece tokenization*) divide palabras poco comunes en sus subpalabras(Rai and Borah 2021). Esta tokenización de subpalabras ayuda al modelo BERT a aprender el significado semántico de palabras relacionadas(Garrido Merchán and Gonzalez Carvajal 2020; Sánchez Mascarell 2021).

La consecuencia de esta tokenización de piezas de palabras y la adición de tokens especiales de BERT es que la longitud de la secuencia después de la tokenización ya no coincide con la longitud de la etiqueta inicial.

Para resolver este problema, necesitamos ajustar la etiqueta de tal manera que tenga la misma longitud que la secuencia después de la tokenización(Boroş et al. 2018):

2.4 Implementación del método propuesto.

Luego de ser descrito el método propuesto, se procede a hacer uso de este en un escenario real y se desarrolla una implementación del método en un prototipo funcional. Se irá describiendo la arquitectura, las tareas, las herramientas, estándares y algoritmos que se aplican en las diferentes fases definidas.

2.4.1 Obtención del dataset.

Los modelos de *Machine Learning* necesitan grandes volúmenes de datos para aprender patrones y posteriormente reconocerlos, para la extracción de entidades nombradas se requieren pares de palabras con su entidad correspondiente. El conjunto de datos de entrenamiento utilizado proviene de CoNLL-2002(Gutiérrez-Fandiño et al. 2021)

Los datos en español son una colección de artículos de noticias publicados por la Agencia de Noticias EFE. Los artículos son de mayo de 2000. La anotación fue realizada por el Centro de Investigación TALP de la Universidad Politécnica de Cataluña (UPC) y el Centro de Lenguaje y Computación (CLiC) de la Universidad de Barcelona (UB), y financiada por la Comisión Europea a través del proyecto NAMIC (IST-1999-12392)(Sang and De Meulder 2003).

2.4.2 Obtención del modelo.

El modelo utilizado está basado en un modelo base *MBERT* ajustado. Ha sido capacitado para reconocer tres tipos de entidades: ubicación (LOC), organizaciones (ORG) y persona (PER).

MBERT es un modelo codificador-decodificador de *Transformer* entrenado en *BERT* y es utilizable con 104 idiomas. Los datos de los 104 idiomas se combinaron cuando se creó *MBERT*. Como resultado, *MBERT* entiende y conoce las relaciones entre las palabras en los 104 idiomas al mismo tiempo.

Este modelo distingue entre el inicio y la continuación de una entidad, de modo que, si hay entidades consecutivas del mismo tipo, el modelo puede generar resultados donde comienza la segunda entidad. Al igual que en el conjunto de datos, cada token se clasificará como una de las siguientes clases:

O	Fuera de una entidad con nombre
B-PER	Comienzo del nombre de una persona justo después del nombre de otra persona
I-PER	Nombre de una persona
B-ORG	Inicio de una organización justo después de otra organización
I-ORG	Organización
B-LOC	Comienzo de una ubicación justo después de otra ubicación
I-LOC	Ubicación

2.4.3 Arquitectura.

La arquitectura que usa el modelo *BERT* básicamente es una pila de codificador de arquitectura de transformador. Una arquitectura de transformador en NLP, resuelve tareas secuencia a secuencia sin los problemas de dependencias largas que presentan las LSTM o

RNN. es una red codificador-decodificador que utiliza la auto atención en el lado del codificador y la atención en el lado del decodificador.

Transformers es una arquitectura novedosa que tiene como objetivo resolver tareas de secuencia a secuencia mientras maneja dependencias de largo alcance con facilidad. Se basa completamente en la auto atención para calcular las representaciones de su entrada y salida sin usar Redes Neuronales Recurrentes alineadas con la secuencia.

Para comprender el funcionamiento de la arquitectura propuesta, en la figura 7 se describe cómo interactúan sus componentes.

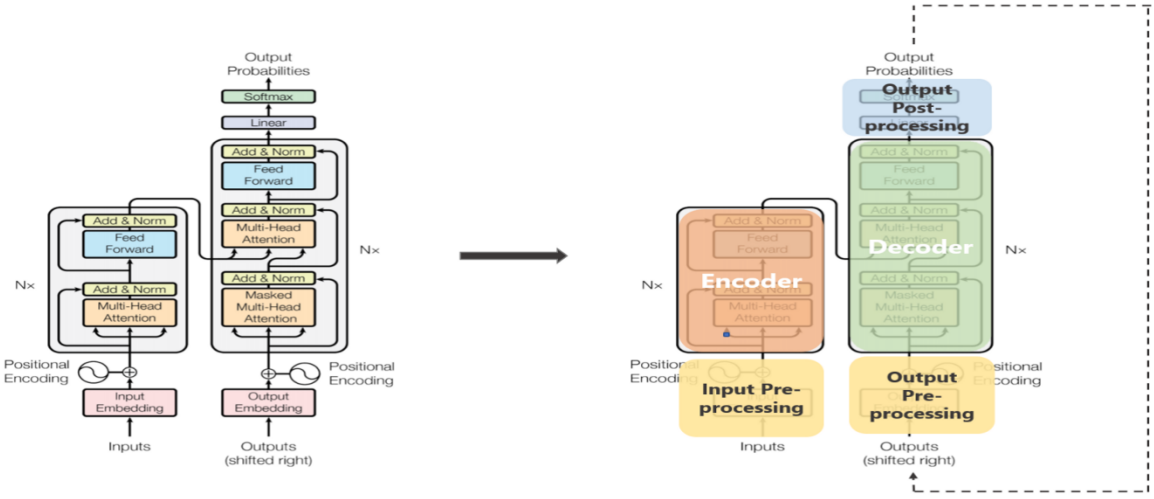


Figura 8 La arquitectura Transformer

Fuente: [Transformers in NLP: A beginner friendly explanation | Towards Data Science]

Las arquitecturas *BERT (BASE y LARGE*) también tienen redes de *feedforward* más grandes (768 y 1024 unidades ocultas respectivamente) y más cabezales de atención (12 y 16 respectivamente) que la arquitectura *Transformer* original. Contiene 512 unidades ocultas y 8 cabezales de atención. BERTBASE contiene 110M parámetros, mientras que *BERTLARGE* tiene 340M parámetros(Acheampong et al. 2021; Koroteev 2021; Rothman 2021).

Este modelo toma el token CLS como entrada primero, luego es seguido por una secuencia de palabras como entrada. Aquí CLS es un token de clasificación. A continuación, pasa la entrada a las capas anteriores. Cada capa aplica auto atención, pasa el resultado a través de una red *feedforward* después de que luego se entrega al siguiente codificador. El modelo produce un vector de tamaño oculto (768 para *BERT BASE*). Si queremos generar un clasificador a partir de este modelo podemos tomar la salida correspondiente al token CLS. Luego este vector entrenado se puede utilizar para realizar una serie de tareas como clasificación, traducción, etc.

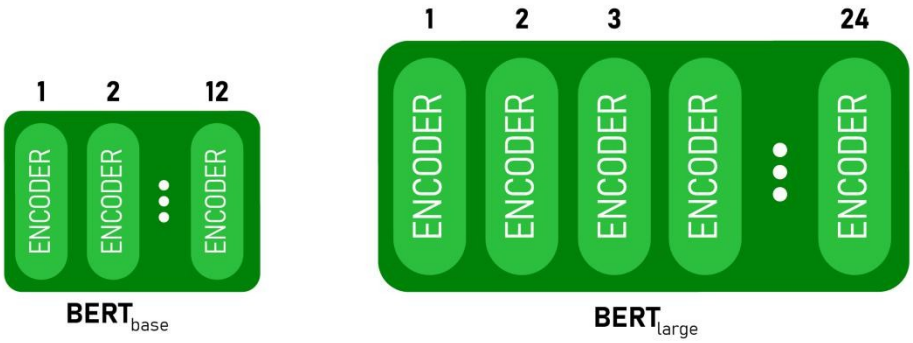


Figura 9 BERT(BASE) and BERT(LARGE).

Fuente Explanation of BERT Model - NLP - GeeksforGeeks

2.4.4 Bibliotecas.

2.4.4.1 *Pandas*.

Pandas es una biblioteca de *Python* con herramientas de análisis de datos. El uso de esta biblioteca le permite manipular datos para obtener información sobre ellos. Wes McKinney creó Pandas y fue desarrollado originalmente para realizar análisis cuantitativos de datos financieros. En 2009, se lanzó Pandas y desde entonces ha ganado popularidad como herramienta para el análisis de datos.

Pandas se usa para el análisis de datos en el campo de la ciencia de datos. La ciencia de datos es simplemente el estudio de datos, con el objetivo de obtener información a partir de conjuntos de datos. Un conjunto de datos podría incluir solo unas pocas entradas o millones

de piezas únicas de información. El objetivo del científico de datos es extraer significado de esos datos a través de un proceso de refinamiento y análisis. Una vez que se ha realizado el análisis, los resultados se pueden visualizar con herramientas como *Matplotlib*, otra biblioteca de *Python*

2.4.4.2 *Spacy*

spaCy es una biblioteca gratuita de código abierto para el procesamiento avanzado del lenguaje natural (NLP) en *Python*.

Características

- Tokenización no destructiva.
- Compatibilidad con tokenización alfa para más de 65 idiomas.
- Soporte integrado para componentes de canalización entrenables, como reconocimiento de entidades nombradas, etiquetado de parte de la voz, análisis de dependencias, clasificación de texto y vinculación de entidades.
- Modelos estadísticos para 17 idiomas.
- Aprendizaje multitarea con transformadores previamente entrenados como *BERT*.
- Compatibilidad con modelos personalizados en *PyTorch* y *TensorFlow*.

2.4.4.3 Streamlit.

Una plataforma de código abierto para que los equipos de aprendizaje automático y ciencia de datos creen aplicaciones de datos con *Python*

La plataforma utiliza *scripts* de *python*, API, implementación instantánea, herramientas de colaboración en equipo y soluciones de administración de aplicaciones para ayudar a los científicos de datos y a los ingenieros de aprendizaje automático a crear aplicaciones basadas en P*ython*.

Las aplicaciones creadas con *Streamlit* van desde aplicaciones capaces de detectar objetos en tiempo real, navegadores de datos geográficos, depuradores de redes de sueños profun-

dos hasta exploradores de GAN facial. Los *frameworks* compatibles con *Streamlit* incluyen: *Scikit Learn, Altair, Bokeh, latex, Keras, Plotly, OpenCV, Vega-Lite, PyTorch, NumPy, Seaborn, Deck.GL, TensorFlow, Python, Matplotlib y Pandas.*

2.4.5 Estándares y tecnologías.

2.4.5.1 JSON

JSON (*JavaScript Object Notation*) es un formato ligero de intercambio de datos. Es fácil para los humanos leer y escribir. Es fácil para las máquinas analizar y generar. *JSON* es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son familiares para los programadores de la familia C de lenguajes, incluyendo C, C ++, C #, *Java*, *JavaScript*, *Perl*, *Python* y muchos otros. Estas propiedades hacen de JSON un lenguaje de intercambio de datos ideal.

JSON se basa en dos estructuras:

Una colección de pares nombre/valor. En varios idiomas, esto se realiza como un objeto, registro, estructura, diccionario, tabla *hash*, lista con claves o matriz asociativa.

Una lista ordenada de valores. En la mayoría de los lenguajes, esto se realiza como una matriz, vector, lista o secuencia.

Conclusiones del capítulo

El modelo propuesto para la extracción de entidades nombradas está basado en fases, siguiendo un enfoque basado en *deep learning*.

La atención al significado semántico de palabras relacionadas utilizada por el método propuesto aporta una solución factible para el problema de extracción de entidades nombradas

Se demostró la viabilidad computacional de la implementación de la herramienta.

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 INTRODUCCIÓN

En el capítulo anterior se explicaron las técnicas utilizadas para diseñar la propuesta de solución. Este capítulo tiene como objetivo la validación de dicha propuesta. Se describen los elementos desarrollados y empleados para realizar la validación. Se explica un componente de *Software* para la extracción de entidades nombradas la cual utiliza los algoritmos seleccionados. También se definen las métricas para la evaluación de la solución. Finalmente se realizan los estudios experimentales diseñados.

3.2 Componente de *Software* para la extracción de entidades nombrada

En la investigación se desarrolló un componente de software para realizar la extracción de entidades nombradas. Para el desarrollo del componente se empleó el lenguaje de programación *Python* en su versión 3.8.8. El componente permite realizar las acciones necesarias para extraer entidades nombradas de los datos de entrada. Primeramente, permite ingresar manualmente los datos a analizar, se realiza el proceso de extracción de entidades y muestra los resultados acompañados de una tabla con información adicional de las entidades, también permite seleccionar que entidades mostrar en los resultados

La Figura 10 muestra la extracción de entidades realizada por el componente sobre un texto dado mostrando las entidades obtenidos.

Figura 10 Aplicación para extracción de entidades nombradas.

3.3 Métricas de evaluación

Para validar la solución desarrollada, se utilizan métricas para evaluar los resultados alcanzados y demostrar su viabilidad. Las más utilizados son **Precisión**, **Recall** y **F- Measure**(Sang and De Meulder 2003; Shelar et al. 2020; Yadav et al. 2020) .

Estos indicadores se utilizan para problemas de recuperación de información(McDonald and Tait 2004). Se utilizan para evaluar los resultados de las búsquedas realizadas en un conjunto de datos en particular. Por otro lado, pueden usarse para problemas de clasificación supervisada para evaluar el rendimiento de los algoritmos de clasificación(Han et al. 2006), y al tener en cuenta la dispersión de los datos analíticos y el equilibrio de los grupos establecidos son adecuados para validar las soluciones propuestas(Han, Kamber and Mining 2006).

El proceso de extracción de entidades nombradas se puede modelar como un problema de clasificación. Se trata de clasificar las entidades en los grupos previamente definidos. Como se menciona en el párrafo anterior las métricas son empleadas en la clasificación

supervisada, esto significa que para cada una de las entidades clasificadas es necesario verificar si en realidad se corresponden con la clasificación recibida. Para realizar esta verificación se desarrolló un trabajo manual donde: se comparó cada una de las entidades clasificados para comprobar la clasificación realizada. Lo anterior permite seleccionar las métricas mencionadas como elementos de validación para la investigación realizada.

A continuación, se explican cada una de las métricas de validación:

3.3.1 Precisión

El término Precisión (P, del inglés *Precision*) es la proporción de los casos predichos positivos que fueron correctos. En el caso particular de la extracción de entidades nombradas, se refiere a la proporción de las entidades clasificadas correctamente. A continuación, se muestra la ecuación que define la métrica:

$$P = \frac{VP}{VP + FP}$$

3.3.2 *Recall*

El término *Recall* (R) es una medida de completitud y representa el porciento de predicciones correctas que fueron etiquetadas como tal. En el caso del problema en cuestión es el porciento de entidades clasificados correctamente. A continuación, se muestra la ecuación que define la métrica:

$$R = \frac{VP}{VP + FN}$$

3.3.3 *F-Measure*

El término *F-Measure* (F) se refiere a una combinación de las métricas Precisión y *Recall*, asignándoles igual peso a ambas. A continuación, se muestra la ecuación que define la métrica:

$$F = \frac{2 * P * R}{P + R}$$

Cada una de las métricas mostradas hacen referencia a términos en las ecuaciones que las definen. A continuación, se describen cada uno de estos:

Verdaderos positivos (VP): entidades que son reconocidas por NER y son clasificadas correctamente.

Falso positivo (FP): entidades que son reconocidas por NER pero que no se clasifican correctamente.

Falso Negativo (FN): entidades dentro del terreno válido que no son reconocidos por NER.

3.4 Resultados experimentales

Se definió un experimento para probar la validez del algoritmo desarrollado, para ello se emplea la aplicación informática presentada anteriormente y las métricas de validación definidas en la sección anterior.

3.4.1 Diseño experimental

Se definieron 3 grupos de noticias diferenciados por el volumen de datos (100, 200 y 300 noticias respectivamente). A continuación, se muestra una tabla con el diseño experimental propuesto.

		Aplicación de la solución
RG_{100}	X_1	O_1
RG_{200}	X_1	O_2
RG_{300}	X_1	O_3

Tabla 2 Diseño experimental propuesto

La simbología utilizada en la tabla anterior es la siguiente:

G_x : Grupo de participantes, el subíndice X representa el grupo de datos conformado, los posibles valores que puede tomar son 100, 200 y 300.

R: Asignación al azar de los participantes en cada uno de los grupos de noticias conformados.

X1 : Tratamiento o estímulo, en este caso la aplicación del algoritmo propuesto.

Ox : Observación realizada luego de la aplicación del algoritmo propuesto.

Seguidamente se aplica la solución propuesta a cada uno de los grupos y se calculan las métricas definidas en las secciones anteriores, facilitando la utilización del diseño experimental propuesto para cada cálculo de las métricas.

El objetivo de las observaciones es detectar la correcta clasificación de las entidades. La observación O1 representa el cálculo de las métricas definidas para el grupo correspondiente al tamaño de 100 noticias. La observación O2 se refiere al cálculo de las métricas para el grupo de 200 noticias. Finalmente, la observación O3 es los cálculos de las métricas para el grupo que posee 300 autores en su registro. Las mediciones se basan en la comparación de los resultados del algoritmo propuesto con su respectivo conjunto de control.

3.4.2. Características de los datos

Los datos fueron obtenidos de los sitios de noticias cubadebate.cu y granma.cu. De Cuba debate se extrajeron noticias de las secciones: Política, economía, ciencia y tecnología, medio ambiente, militar e inteligencia y sociedad. En el caso del Granma las noticias pertenecen a las secciones: Cuba, mundo, cultura, deportes, ciencia y salud. La selección de los registros bibliográficos se basó en el método de muestreo aleatorio simple inicialmente, garantizando la aleatoriedad de los datos y la representatividad de las muestras seleccionadas.

3.4.3. Análisis de los resultados

De acuerdo al diseño experimental propuesto se realizaron varias pruebas a cada uno de los grupos conformados encaminadas al cálculo de las métricas definidas. La primera métrica

calculada en cada uno de los grupos de autores formados y sus respectivos conjuntos de datos fue **Precisión**.

Los resultados obtenidos muestran variaciones entre los grupos experimentales de 100 noticias y el resto. También se puede apreciar que apenas existen variaciones entre los grupos de 200 y 300 autores. El análisis anterior permite concluir que la solución se desempeña con mayor precisión cuando el volumen de datos es elevado.

La tabla siguiente muestra los resultados obtenidos en los cálculos de la precisión con los grupos de pruebas conformados.

Cantidad de noticias	Nivel de Precisión
100	0.91
200	0.95
300	0.97

Tabla 3 Resultados de la métrica precisión

A partir de los datos obtenidos en las pruebas realizadas para la métrica precisión se aplicó la medida de tendencia central (media) y se calculó su desviación estándar. La desviación estándar de una serie de mediciones es el promedio de desviación de cada una de las mediciones con respecto a la media de estas. Cuanto mayor es la dispersión de los datos con respecto a la media mayor es la desviación estándar. A continuación, se muestra la ecuación que define el promedio mencionado.

$$S = \sqrt{\frac{\sum (X - Xm)^2}{N}}$$

En la ecuación anterior la variable X representa los valores de las mediciones realizadas, la variable Xm representa la media de las mediciones obtenidas. La variable N representa la cantidad de mediciones realizadas.

Para el cálculo de la desviación estándar se sigue el procedimiento descrito a continuación:

1. Se ordenan las mediciones.

2. Se calcula la media de las mediciones realizadas.

3. Se determina la desviación de cada medición con respecto a la media.

4. Se eleva al cuadrado cada desviación y se obtiene la sumatoria de las desviaciones elevadas al cuadrado o P(X − Xm) 2 .

5. Se aplica la formula con los valores obtenidos.

A partir de lo anterior se obtiene un valor medio de la precisión de **0.94** o **94%** con una desviación estándar de **0.02**.

La métrica *Recall* está condicionada por las clasificaciones de las entidades. La solución desarrollada en ninguna de las pruebas realizadas ignoro la clasificación de ninguna entidad propiciando la obtención de un 100 % para esta métrica.

Finalmente se calculó la métrica *F-Measure*. El grupo de noticias con 100 registros es el más afectado con un valor medio de 94 % mientras que para los grupos de 200 y 300 autores se obtuvo un 96 % y 97 % respectivamente. En el caso de los conjuntos de datos con un nivel de ambigüedad bajo (25 % de los autores) el valor medio obtenido fue de 93 % mientras que para los conjuntos de datos de medio y alto nivel de ambigüedad se obtuvo un valor de 98 % y 96 % respectivamente. A partir de lo anterior se puede concluir que la solución propuesta obtiene mejores resultados para la métrica *F-Measure* cuando el volumen de datos es elevado y el nivel de ambigüedad también.

A continuación, se muestra una tabla donde se resumen los resultados obtenidos en el cálculo de la métrica *F-Measure*.

Cantidad de noticias	*F-Measure*
100	0.95

200	0.97
300	0.98

Con los resultados mostrados se obtiene un valor medio para la métrica *F-Measure* de **0.96** o **96 %** mientras que su desviación estándar es de **0.01**.

Los resultados obtenidos reflejan las condiciones en que la solución propuesta se comporta de mejor forma, así como los casos peores. En todas las métricas calculadas un alto volumen de datos arrojó mejores resultados.

La métrica precisión establece la proporción de las entidades clasificados correctamente. Teniendo un 94 % de precisión en los resultados obtenidos por el algoritmo, se puede afirmar que: de cada 100 entidades clasificadas, 94 son clasificados correctamente.

Conclusiones del capítulo

El análisis realizado permite afirmar que la solución desarrollada obtiene mejores resultados cuando el volumen de datos es elevado.

Los resultados obtenidos de las métricas Precisión y *Recall* permitieron evaluar el desempeño de la herramienta para la extracción de entidades nombradas.

La aplicación demostró dar solución al problema de extracción de entidades nombradas.

CONCLUSIONES

En la presente investigación se plantearon una serie de objetivos los cuales se fueron cumpliendo progresivamente, permitiendo arribar a las siguientes conclusiones:

Partiendo del estudio de los campos del Procesamiento de Lenguaje y el *Machine Learning* se identificaron los enfoques y características necesarios para dar solución a la problemática.

La propuesta de un modelo y su posterior implementación permitió definir las funcionalidades de una aplicación informática para cumplir objetivos específicos.

A partir de la aplicación de la propuesta de solución al caso de estudio se pudo constatar que se resuelve el problema de extracción de entidades nombradas de artículos de prensa en español.

RECOMENDACIONES

A partir del estudio realizado en la presente investigación se propone continuar agregándole funcionalidades a la solución con el objetivo de mejorar la calidad de los procesos que realiza como parte de su funcionamiento interno.

REFERENCIAS BIBLIOGRÁFICAS

ACHEAMPONG, F. A., H. NUNOO-MENSAH AND W. J. A. I. R. CHEN Transformer models for text-based emotion detection: a review of BERT-based approaches 2021, 54(8), 5789-5829.

AKBIK, A., T. BERGMANN AND R. VOLLGRAF. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* 2019, p. 724-728.

ALBARED, M., M. G. OCAÑA, A. GHAREB AND T. AL-MOSLMI. Recent progress of named entity recognition over the most popular datasets. In *2019 First International Conference of Intelligent Computing and Engineering (ICOICE).* IEEE, 2019, p. 1-9.

AZEVEDO, A. AND M. F. J. I.-D. SANTOS KDD, SEMMA and CRISP-DM: a parallel overview 2008.

BALOG, K. *Entity-oriented search*. Edtion ed.: Springer Nature, 2018. ISBN 3319939351.

BOROŞ, T., Ş. D. DUMITRESCU AND R. BURTICA. NLP-Cube: End-to-end raw text processing with neural networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.* 2018, p. 171-179.

CHAUDHARY, A., J. XIE, Z. SHEIKH, G. NEUBIG, et al. A little annotation does a lot of good: A study in bootstrapping low-resource named entity recognizers 2019.

CHHABRA, A., P. BRANCO, G.-V. JOURDAN AND H. L. VIKTOR. An extensive comparison of systems for entity extraction from log files. In *International Symposium on Foundations and Practice of Security.* Springer, 2022, p. 376-392.

CHIONG, R. AND W. WEI. Named entity recognition using hybrid machine learning approach. In *2006 5th IEEE International Conference on Cognitive Informatics.* IEEE, 2006, vol. 1, p. 578-583.

DENG, L., D. J. F. YU AND T. I. S. PROCESSING Deep learning: methods and applications 2014, 7(3–4), 197-387.

DODDINGTON, G. R., A. MITCHELL, M. A. PRZYBOCKI, L. A. RAMSHAW, et al. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec.* Lisbon, 2004, vol. 2, p. 837-840.

ELMAN, J. L. J. C. S. Finding structure in time 1990, 14(2), 179-211.

FALCK, F., J. MARSTALLER, N. STOEHR, S. MAUCHER, et al. Measuring proximity between newspapers and political parties: the sentiment political compass 2020, 12(3), 367-399.

FIGUEROLA, C. G., M. J. D. S. ESCOBAR AND S. R. B. O. ABSTRACTS DIGITAL NEWS PRESS MINING THROUGH TOPIC MODELING, ENTITY RECOGNITION AND SOCIAL NETWORKS ANALYSIS TECHNIQUES, 63.

GAMBOA-ROSALES, N.-K., A. CASTORENA-ROBLES, M.-A. CASAS-VALADEZ, M.-J. COBO, et al. Decision Making using Internet of Things and Machine Learning: A bibliometric approach to tracking main research themes. In *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI).* IEEE, 2020, p. 1-6.

GARRIDO MERCHÁN, E. C. AND S. GONZALEZ CARVAJAL Comparing BERT against traditional machine learning text classification 2020.

GAYTÁN DÍAZ, C. R. Método de aprendizaje probabilístico para el reconocimento de entidades nombradas. Tesis (MC)--Centro de Investigación y de Estudios Avanzados del IPN Unidad …, 2018.

GELBUKH, A. J. K. S. Procesamiento de lenguaje natural y sus aplicaciones 2010, 1, 6-11.

GOYAL, A., V. GUPTA AND M. J. C. S. R. KUMAR Recent named entity recognition and classification techniques: a systematic review 2018, 29, 21-43.

GRISHMAN, R. AND B. M. SUNDHEIM. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics.* 1996.
GULLI, A. AND S. PAL *Deep learning with Keras*. Edtion ed.: Packt Publishing Ltd, 2017. ISBN 1787129039.
GUTIÉRREZ-FANDIÑO, A., J. ARMENGOL-ESTAPÉ, M. PÀMIES, J. LLOP-PALAO, et al. Spanish language models 2021.
HAN, J., M. KAMBER AND D. J. M. K. MINING Concepts and techniques 2006, 340, 94104-93205.
HAYKIN, S. *Neural networks and learning machines, 3/E*. Edtion ed.: Pearson Education India, 2009. ISBN 933258625X.
HERNÁNDEZ, C. AND J. E. R. J. R. V. RODRÍGUEZ Preprocesamiento de datos estructurados 2008, 4(2), 27-48.
HOLLENSTEIN, N. AND C. J. A. P. A. ZHANG Entity recognition at first sight: Improving NER with eye movement information 2019.
KETKAR, N. AND J. MOOLAYIL. Introduction to pytorch. In *Deep learning with python.* Springer, 2021, p. 27-91.
KHADIR, A. C., H. ALIANE AND A. J. C. S. R. GUESSOUM Ontology learning: Grand tour and challenges 2021, 39, 100339.
KOROTEEV, M. J. A. P. A. BERT: A review of applications in natural language processing and understanding 2021.
KRIPKE, S. A. Naming and necessity. In *Semantics of natural language.* Springer, 1972, p. 253-355.
LI, J., A. SUN, J. HAN, C. J. I. T. O. K. LI, et al. A survey on deep learning for named entity recognition 2020, 34(1), 50-70.
LIU, M., Z. TU, Z. WANG AND X. J. A. P. A. XU LTP: a new active learning strategy for BERT-CRF based named entity recognition 2020.
LIU, T., J.-G. YAO AND C.-Y. LIN. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* 2019, p. 5301-5307.
MANASWI, N. K. Understanding and working with Keras. In *Deep Learning with Applications Using Python.* Springer, 2018, p. 31-43.
MANSOURI, A., L. S. AFFENDEY, A. J. I. J. O. C. S. MAMAT AND N. SECURITY Named entity recognition approaches 2008, 8(2), 339-344.
MCDONALD, S. AND J. TAIT *Advances in Information Retrieval: 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings*. Edtion ed.: Springer, 2004. ISBN 3540247521.
MI, W., Y. LI AND S. WANG. Empirical evaluation of the active learning strategies on software defects prediction. In *2020 6th International Symposium on System and Software Reliability (ISSSR).* IEEE, 2020, p. 83-89.
MISSAOUI, S., A. MACFARLANE, S. MAKRI AND M. GUTIERREZ-LOPEZ. DMINR at TREC News Track. In *TREC.* 2019.
MOREIRA, D., I. CRUZ, K. GONZALEZ, A. QUIRUMBAY, et al. Análisis del Estado Actual de Procesamiento de Lenguaje Natural 2021, (E42), 126-136.
NADEAU, D. AND S. J. L. I. SEKINE A survey of named entity recognition and classification 2007, 30(1), 3-26.
NASAR, Z., S. W. JAFFRY AND M. K. J. A. C. S. MALIK Named entity recognition and relation extraction: State-of-the-art 2021, 54(1), 1-39.

NIVRE, J. On statistical methods in natural language processing. In *Proceedings of the 13th Nordic Conference of Computational Linguistics (NODALIDA 2001).* 2001.
PASCUAL, C. P. J. O. En defensa del procesamiento del lenguaje natural fundamentado en la lingüística teórica 2012, (26), 13-48.
PETASIS, G., A. CUCCHIARELLI, P. VELARDI, G. PALIOURAS, et al. Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval.* 2000, p. 128-135.
RAI, A. AND S. BORAH. Study of various methods for tokenization. In *Applications of Internet of Things.* Springer, 2021, p. 193-200.
RIJHWANI, S., S. ZHOU, G. NEUBIG AND J. J. A. P. A. CARBONELL Soft gazetteers for low-resource named entity recognition 2020.
ROTHMAN, D. *Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more*. Edtion ed.: Packt Publishing Ltd, 2021. ISBN 1800568630.
RUSNACHENKO, N. AND N. LOUKACHEVITCH. Neural network approach for extracting aggregated opinions from analytical articles. In *International Conference on Data Analytics and Management in Data Intensive Domains.* Springer, 2018, p. 167-179.
SÁNCHEZ MASCARELL, M. Clasificación de textos basado en los modelos pre-entrenados BERT. Universitat Politècnica de València, 2021.
SANG, E. F. AND F. J. A. P. C. DE MEULDER Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition 2003.
SHAFFER, K. Language Clustering for Multilingual Named Entity Recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2021, p. 40-45.
SHARNAGAT, R. J. C. F. I. L. T. Named entity recognition: A literature survey 2014, 1-27.
SHELAR, H., G. KAUR, N. HEDA, P. J. S. AGRAWAL, et al. Named entity recognition approaches and their comparison for custom ner model 2020, 39(3), 324-337.
SINTAYEHU, H. AND G. J. I. J. O. I. T. LEHAL Named entity recognition: a semi-supervised learning approach 2021, 13(4), 1659-1665.
SONG, X., A. SALCIANU, Y. SONG, D. DOPSON, et al. Fast wordpiece tokenization 2020.
SRIHARI, R. K. A hybrid approach for named entity and sub-type tagging. In *Sixth Applied Natural Language Processing Conference.* 2000, p. 247-254.
SUN, P., X. YANG, X. ZHAO AND Z. WANG. An overview of named entity recognition. In *2018 International Conference on Asian Language Processing (IALP).* IEEE, 2018, p. 273-278.
TORIBIO, R., P. MARTÍNEZ AND C. J. P. D. L. N. DE PABLO-SÁNCHEZ Evaluación de la Extracción de Entidades Nombradas de OpenCalais en castellano 2010, (45), 287-290.
TORRES, M. M. E. AND R. J. R. P. MANJARRÉS-BETANCUR Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural 2020, 16(31), 85-96.
VASILEV, I., D. SLATER, G. SPACAGNA, P. ROELANTS, et al. *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*. Edtion ed.: Packt Publishing Ltd, 2019. ISBN 1789349702.
VÁSQUEZ, A. C., J. P. QUISPE AND A. M. J. R. D. I. D. S. E. I. HUAYNA Procesamiento de lenguaje natural 2009, 6(2), 45-54.
WANG, J., L. SHOU, K. CHEN AND G. CHEN. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 2020, p. 5918-5928.

WANG, Q. AND X. J. A. S. SU Research on Named Entity Recognition Methods in Chinese Forest Disease Texts 2022, 12(8), 3885.
YADAV, H., S. GHOSH, Y. YU AND R. R. J. A. P. A. SHAH End-to-end named entity recognition from english speech 2020.
YADAV, V. AND S. J. A. P. A. BETHARD A survey on recent advances in named entity recognition from deep learning models 2019.