EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación y Desarrollo Tecnológico

Tesis de Doctorado

Minería de referencias bibliográficas: Mejora en la generalización de la Segmentación

presentada por
M.C.C. Rodrigo Cuéllar Hidalgo

como requisito para la obtención del grado de
Doctor en Ciencias de la Computación

Director de tesis
Dr. Raúl Pinto Elías

Codirector de tesis
Dr. Gerardo Reyes Salgado
Dr. Juan Manuel Torres Moreno

Cuernavaca, Morelos, México. Diciembre de 2024.

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

cenidet Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL	Código: CENIDET-AC-006-D20
		Revisión: O
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4	Página 1 de 1

Cuernavaca, Mor., a 31 de octubre de 2024

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

AT´N: DR. JUAN GABRIEL GONZÁLEZ SERNA
PRESIDENTE DEL CLAUSTRO DOCTORAL
DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Los abajo firmantes, miembros del Comité Tutorial del estudiante **M.C. RODRIGO CUÉLLAR HIDALGO** manifiestan que después de haber revisado el documento de tesis titulado: **"Minería de referencias bibliográficas: Mejora en la generalización de la Segmentación"**, realizado bajo la dirección del Dr. Raúl Pinto Elías y la codirección del Dr. Juan Manuel Torres Moreno y del Dr. Gerardo Reyes Salgado el trabajo se ACEPTA para proceder a su impresión.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

Dr. Raúl Pinto Elías
TecNM/CENIDET

DR. GERARDO REYES SALGADO
UNIVERSIDAD REY JUAN CARLOS,ESPAÑA

MANUEL 3

DR. JUAN MANUEL TORRES MORENO
UNIVERSIDAD DE AVIGNON,FRANCIA

DRA. ANDREA MAGADÁN SALAZAR
TecNM/CENIDET

DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
TecNM/CENIDET

DR. NIMROD GONZÁLEZ FRANCO
TecNM/CENIDET

Osslan Osiris Vergara Villegas

DR. OSSLAN OSIRIS VERGARA VILLEGAS
UNIVERSIDAD AUTONOMÁ DE CIUDAD JUÁREZ

c.c.p: C María Elena Gómez Torres/Jefa del Departamento de Servicios Escolares
c.c.p: C Noé Alejandro Castro Sánchez/Jefe del Departamento de Ciencias Computacionales
c.c.p: Expediente

TECNOLÓGICO
NACIONAL DE MÉXICO

Centro Nacional de Investigación
y Desarrollo Tecnológico
Subdirección Académica

Cuernavaca, Mor., 05/noviembre/2024
No. De Oficio: SAC/350/2024
Asunto: Autorización de impresión de tesis

RODRIGO CUÉLLAR HIDALGO
CANDIDATO AL GRADO DE DOCTOR
EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Minería de referencias bibliográficas: Mejora en la generalización de la Segmentación"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica·
"Conocimiento y Tecnología al Servicio de México"

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz

Agradecimientos

Al Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET), por brindarme sus instalaciones y recursos para el desarrollo de este trabajo.

A El Colegio de México, por las facilidades otorgadas y, en especial, a la Mtra. Micaela Chávez, directora de la Biblioteca Daniel Cosío Villegas, por su apoyo constante y paciencia.

A mi primer director, el Dr. Gerardo Reyes Salgado de la Universidad Rey Juan Carlos en España, quien creyó en mí desde el inicio y me dio la oportunidad de entrar al programa. Su confianza y guía fueron cruciales en esta etapa inicial.

A mi director de tesis, el Dr. Raúl Pinto Elías, quien asumió el rol de director en la etapa final de este trabajo, brindándome orientación constante, consejos invaluables, y, especialmente, su paciencia para llevar a cabo esta tesis.

A mi codirector de tesis, el Dr. Juan Manuel Torres-Moreno de la Universidad de Aviñón en Francia, por su asesoría y apoyo en cada paso del camino.

A mis revisores, la Dra. Andrea Magadán, el Dr. Noé Castro y el Dr. Nimrod González, cuyos comentarios, críticas y ánimos fueron esenciales para la realización de esta tesis.

A mis compañeros y amigos del Departamento de Ciencias Computacionales, quienes me brindaron consejos, información y su valiosa compañía. En especial, al grupo del Seminario de Inteligencia Artificial, con quienes compartí grandes momentos.

A mi amada esposa, Kenia Martínez Tinajero, por su infinita paciencia para soportar las horas que pasé lejos de ella, su afecto, comprensión incondicional y completo apoyo, sin los cuales esta investigación no hubiera podido llevarse a cabo.

A mis padres, Carlos Francisco Cuéllar Aguilar y Gilda Odelisa Hidalgo Bolaños, por darme la vida, sus cuidados y su apoyo incondicional a lo largo de esta trayectoria.

A mis hijas, Amy y Jenny, quienes junto a mi esposa han sido el motor de mi vida y mi mayor fuente de motivación para alcanzar esta meta.

Resumen

La tesis se centra en mejorar la segmentación de referencias bibliográficas mediante el uso de aprendizaje automático y arquitecturas de redes neuronales. El objetivo principal es desarrollar y evaluar un modelo que realice la segmentación de referencias en múltiples idiomas y estilos bibliográficos. Se evaluaron varios enfoques, incluyendo arquitecturas como CRF, BiLSTM, Transformer y Ventanas Deslizantes, así como sus combinaciones, asegurando la resiliencia frente a errores y variaciones en los datos de entrada.

For this purpose, a multilingual corpus of bibliographic references was developed, encompassing a range of styles and marking a significant advancement in scientific knowledge. This approach is particularly pertinent for digital libraries in non-English-speaking countries, where access to robust reference processing tools remains limited.

El proceso de segmentación se abordó en varias fases, comenzando con la selección manual y automática de características, seguida de la captura de contexto mediante arquitecturas como BiLSTM y Transformer. Las predicciones se basaron en estas técnicas, priorizando la tolerancia a omisiones e inconsistencias en la segmentación.

Los experimentos mostraron que las combinaciones de BiLSTM y Transformer lograron más del 98 % de F-score en segmentación de referencias y más del 92 % en entornos multilingües. Cabe destacar que Transformer + BiLSTM y Ventanas Deslizantes + BiLSTM se destacaron por su eficiencia y alto rendimiento en condiciones desafiantes.

La tesis enfatiza la importancia crítica de la selección de características y la complejidad computacional. A pesar de la mayor eficiencia de los modelos combinados, requieren recursos computacionales significativos, lo cual presenta una limitación para su aplicación práctica.

En conclusión, el estudio proporciona un marco sólido para la segmentación de referencias en múltiples idiomas y estilos. Destaca la efectividad de combinar BiLSTM y Transformer para lograr precisión y robustez frente a errores. Además, sienta las bases para futuras investigaciones que ampliarán la generalización a más idiomas y estilos y optimizarán la eficiencia computacional.

Abstract

The thesis focuses on improving bibliographic reference segmentation through the use of machine learning and neural network architectures. The primary objective is to develop and evaluate a model that performs reference segmentation across various languages and bibliographic styles. Multiple approaches are evaluated, including architectures such as CRF, BiLSTM, Transformer, and Sliding Window, as well as their combinations, ensuring resilience to errors and variations in the input data.

To achieve this objective, a multilingual corpus of bibliographic references was developed, covering diverse styles, representing a step towards scientific knowledge. This approach is particularly relevant for digital libraries in non-English-speaking countries, where access to robust reference processing tools is limited.

The segmentation process was addressed in several phases, starting with the manual and automatic selection of features, followed by context capturing using architectures such as BiLSTM and Transformer. Predictions were based on these techniques, prioritizing tolerance to omissions and inconsistencies in segmentation.

Experiments showed that combinations of BiLSTM and Transformer achieved over 98 % F-score in reference segmentation and more than 92 % in multilingual environments. Notably, Transformer + BiLSTM and Sliding Window + BiLSTM stood out for their efficiency and high performance under challenging conditions.

The thesis emphasizes the critical importance of feature selection and computational complexity. Despite the higher efficiency of combined models, they require significant computational resources, which presents a limitation for practical application.

In conclusion, the study provides a solid framework for reference segmentation in multiple languages and styles. It highlights the effectiveness of combining BiLSTM and Transformer to achieve precision and robustness against errors. Additionally, it lays the groundwork for future research that will expand generalization to more languages and styles and optimize computational efficiency.

Índice general

Índice de figuras

Índice de tablas

Lista de acrónimos

ANN - Artificial Neural Network

Bi-LSTM - Bidirectional Long Short Term Memory

CNN - Convolutional Neural Network

CRF - Conditional Random Field

DL - Deep Learning

FBA - Frame Based Approach

HMM - Hidden Markov Models

KB - Knowledge Base

LSTM - Long Short Term Memory

ML - Machine Learning

NLP - Natural Language Processing

OCR - Optical Character Recognition

PBA - Principle Based Approach

RF - Random Forest

BPE - Byte Pair Encoding

1 Introducción

Las bibliotecas digitales apoyan la investigación académica mediante el acceso a volúmenes amplios de información. Además de preservar datos, organizan y agrupan recursos académicos, como artículos, libros y datos, para facilitar su recuperación. Estas funciones de organización y agrupación son fundamentales para desarrollar herramientas avanzadas, como la visualización de citas y la creación de redes de colaboración entre autores, las cuales amplían las posibilidades de recuperación de información y análisis. Estas herramientas permiten a los investigadores navegar eficazmente la creciente literatura académica, siendo esenciales en un contexto de expansión constante de publicaciones. (Morán Reyes, 2020)

Uno de los procesos esenciales para el desarrollo y mejora de los servicios dentro de una biblioteca digital, es la Minería de referencias bibliográficas (Pena-Rocha y cols., 2024).

La Minería de referencias consiste en detectar, extraer y segmentar correctamente las referencias contenidas en los documentos académicos. Las referencias, bien segmentadas, permiten realizar tareas como la evaluación del impacto de un investigador, el análisis de similitud entre documentos o la recomendación automática de artículos relevantes. Sin embargo, a pesar de los avances en la automatización de los procesos antes mencionados, persisten limitaciones significativas (Tkaczyk, Gupta, y cols., 2018).

El problema es que los sistemas actuales de Minería de referencias no han logrado generalizar a diferentes lenguas y estilos de citación. Derivando en una precisión limitada, especialmente en documentos que no están en inglés (Jain y cols., 2023). Además, errores como la mala digitalización, la ambigüedad en los nombres de autores y las inconsistencias en la estructura de las referencias agravan la situación. Afectando la calidad de la extracción de datos (Patro, 2012; Tkaczyk y cols., 2015; Agrawal y cols., 2019).

> *"La Minería de referencias puede verse como invertir el proceso de formatear un registro bibliográfico en una cadena. Durante el formateo se pierde parte de la información y, por lo tanto, el proceso inverso no es una tarea trivial y, por lo general, introduce errores. Hay algunos desafíos relacionados con el análisis de referencias."*
>
> Tkaczyk, Gupta, y cols. (2018)

Entre los principales desafíos a los que se enfrenta la Minería de referencias están los siguientes:

- La dificultad de identificar el tipo de documento al que se refiere la referencia, lo que impide extraer correctamente los metadatos asociados.

- La variedad de estilos de citación, lo que complica la localización precisa de los componentes de la referencia (autor, título, año, etc.).
- Los errores humanos en la escritura y las imprecisiones durante la digitalización, que incluyen fallos en el reconocimiento óptico de caracteres (OCR), omisiones o excesos de espacios, y errores en la puntuación o en el uso de delimitadores.

La mayoría de los avances en la Minería de referencias han estado enfocados en corpus en inglés, particularmente en los dominios de las ciencias de la computación y la salud. Limitando su generalización en otros contextos lingüísticos y científicos, creando una brecha en la capacidad las herramientas actuales.

Mejorar la generalización de la segmentación de referencias a otros idiomas y estilos de citación es fundamental para optimizar la eficacia de los sistemas de recuperación de información en repositorios científicos multilingües y multidisciplinarios (Santos y cols., 2023).

1.1. Planteamiento del problema

El estado del arte en la Minería de referencias (RM) presenta limitaciones significativas que dificultan la generalización de sus aplicaciones. Los principales problemas identificados en el área incluyen:

1. **Dependencia casi exclusiva del idioma inglés**: La mayoría de los modelos de RM se han desarrollado y probado principalmente en corpus en inglés. Esta focalización limita su rendimiento cuando se aplican a referencias en otros idiomas, reduciendo su efectividad en contextos multilingües.
2. **Sesgo hacia estilos de citación de ciencias de la computación y ciencias de la salud**: Los corpus de entrenamiento y validación de los modelos de RM suelen provenir en su mayoría de las ciencias de la computación y de la salud, lo cual restringe su habilidad para gestionar referencias de otros campos del conocimiento. Estos otros campos emplean estructuras y estilos de citación que pueden diferir considerablemente.

Las limitaciones, mencionadas anteriormente, generan un sesgo hacia dominios específicos y una exclusión de idiomas distintos al inglés, lo que disminuye la precisión y aplicabilidad de los modelos en un entorno diverso de referencias bibliográficas.

El presente trabajo busca mitigar estas limitaciones proponiendo un modelo diseñado para segmentar referencias bibliográficas en múltiples idiomas y estilos, ampliando su aplicabilidad a una variedad más extensa de contextos lingüísticos y disciplinarios.

Solamente una publicación (Boukhers y cols., 2019) ha mencionado la necesidad de explorar técnicas de RM en otros idiomas. Los únicos trabajos relevantes incluyen los de Körner y cols. (2017) en alemán, y algunos intentos multilingües como:

- Prasad y cols. (2018), quienes traducen referencias al inglés como un paso intermedio.

- Rodrigues Alves y cols. (2018), quienes entrenan modelos con referencias en inglés, francés, alemán, español y latín.
- Choi y cols. (2023), quienes emplean un modelo multilingüe basado en BERT.

Dichas limitaciones coinciden con lo expresado por Jain y cols. (2023), quienes señalan que los sesgos hacia ciertos idiomas y estilos restringen la generalización de la RM. Además, la diversidad en la implementación de modelos y la variación en los corpus de entrenamiento dificultan la comparación entre propuestas descritas en la literatura.

1.1.1. Delimitación del problema específico

Este trabajo e centra en desarrollar un modelo para la subtarea de segmentación en la Minería de referencias, aplicable en diferentes idiomas y estilos bibliográficos, mejorando la generalización en diversos dominios del conocimiento.

1.1.2. Complejidad del problema

La RM enfrenta varios desafíos:

- Errores humanos en la entrada de datos, lo que provoca incumplimientos en los formatos de citación.
- Fallos en los procesos de digitalización, como errores en el reconocimiento óptico de caracteres (OCR).
- Similitudes y omisiones ambiguas en diferentes campos de las referencias.

Además, Bender (2019) advierte que la variabilidad lingüística entre idiomas afecta negativamente la efectividad de los modelos de aprendizaje automático, ya que suelen ser diseñados y probados solamente en corpus en inglés. Limitando su generalización en otros contextos.

Para abordar los mencionados desafíos, se plantean los siguientes subproblemas:

- Crear corpus para entrenamiento y evaluación en al menos tres idiomas.
- Desarrollar modelos capaces de segmentar referencias en diferentes idiomas y estilos.
- Evaluar diversas arquitecturas para identificar cuál es más tolerante a errores.

1.2. Propuesta de solución

1.2.1. Objetivo general

Construir un modelo que permita segmentar referencias en diferentes idiomas y estilos bibliográficos.

1.2.2. Objetivos específicos

- Analizar el estado del arte en Minería de referencias para identificar los enfoques predominantes.
- Crear un corpus multilingüe y con diversos estilos bibliográficos.
- Determinar la arquitectura más tolerante a errores en la segmentación de referencias.
- Establecer una comparación con el estado del arte que mantenga condiciones uniformes, mediante la recreación de la arquitectura mas representativa.

1.2.3. Alcances

El modelo desarrollado puede segmentar referencias bibliográficas en tres idiomas diferentes. Para optimizar su desempeño, se comparan diversas arquitecturas de modelos, con el fin de identificar aquella con mayor tolerancia a errores. Además, las pruebas se realizan en dos corpus diferentes del utilizado para el entrenamiento, lo que permite evaluar y validar la capacidad de generalización del modelo.

1.2.4. Limitaciones

- El sistema solamente realiza la subtarea de segmentación en la Minería de referencias.
- Para la evaluación de tolerancia a errores se utiliza el corpus CORA, pero no se contempla ningún tipo de cuantificación de errores en el mismo.
- El entrenamiento y las pruebas de los modelos dependen de los recursos de cómputo disponibles.
- Los recursos computacionales no permiten el uso de recursos más avanzados, como es el caso de BERT.
- La comparación con el estado del arte no posible debido a lo mencionado por Jain y cols. (2023)

1.3. Justificación

Dado el aumento continuo de la literatura científica, es fundamental desarrollar herramientas que apoyen a las bibliotecas digitales en la gestión y análisis bibliométricos (Santos y cols., 2023).

La automatización de procesos como la segmentación de referencias bibliográficas permite optimizar el manejo de grandes volúmenes de información, facilitando la organización, búsqueda y recuperación eficiente de datos. Lo que resulta esencial en un contexto de expansión acelerada de publicaciones académicas, donde la sobrecarga de información representa un desafío creciente para investigadores y bibliotecarios.

Mejorar la segmentación de referencias en idiomas diferentes al inglés y en una variedad de estilos de citación contribuirá a la democratización del conocimiento científico, asegurando que las bibliotecas digitales puedan procesar material académico de manera inclusiva y eficiente.

Dado que la mayor parte de las herramientas existentes se desarrollan con un enfoque en inglés y en dominios específicos como las ciencias de la salud y la computación, extender estas capacidades a otros contextos lingüísticos y disciplinarios permitirá reducir los sesgos existentes en el procesamiento de referencias.

Esto es particularmente relevante para bibliotecas en países no angloparlantes, que enfrentan limitaciones en el acceso a herramientas tecnológicas especializadas.

Contar con sistemas que soporten la segmentación en múltiples idiomas y estilos de citación favorecerá la interoperabilidad entre repositorios digitales y permitirá una mejor integración de datos entre instituciones. A su vez, esto coadyuva a cumplir con las funciones fundamentales de las bibliotecas, como la organización, preservación y difusión del conocimiento, fortaleciendo su papel en la gestión de recursos bibliográficos en un entorno académico cada vez más globalizado.

1.4. Beneficios

La ampliación de las capacidades de la Minería de referencias (RM) tiene un impacto significativo en la comunidad académica global, especialmente para investigadores y bibliotecas digitales de países no angloparlantes y de disciplinas con menor acceso a herramientas tecnológicas avanzadas (Morán Reyes, 2020; Santos y cols., 2023; Pena-Rocha y cols., 2024). Los beneficios de este desarrollo incluyen:

- Ampliar el acceso a herramientas de RM permitirá a investigadores de países no angloparlantes, y de disciplinas menos tecnificadas, mejorar su gestión de referencias y revisiones bibliográficas.
- Incrementará la precisión de los servicios de búsqueda y recuperación de información en bibliotecas digitales no angloparlantes.
- Fomentará la automatización en los procesos de evaluación de la calidad científica fuera del mundo angloparlante.
- Detectar, segmentar y validar referencias correctamente es determinante para el análisis bibliométrico y la identificación de tendencias científicas.

1.5. Estructura del documento

El documento continúa con el Capítulo 2, donde se presenta el marco teórico y el análisis del estado del arte. El Capítulo 3 describe la propuesta de investigación para mejorar los vacíos detectados y detalla la construcción del modelo y los conjuntos de datos. El Capítulo 4 expone los experimentos realizados para evaluar el cumplimiento de los objetivos. Finalmente, el Capítulo 5 presenta las conclusiones, el trabajo futuro y las contribuciones logradas.

2 Trabajos relacionados

En el presente capítulo se describen los conceptos teóricos empleados para el desarrollo de este proyecto doctoral. De igual manera, se analiza el estado del arte donde se describen los diversos abordajes para resolver la tarea de RM, y sus diferentes subtareas.

2.1. Marco Teórico

2.1.1. Ciencias de la información

Es el área de conocimiento comprende el análisis, recolección, clasificación, uso, almacenamiento, recuperación, difusión y protección de la información.

A partir de las ciencias de la información se recuperan los siguientes conceptos:

Texto académico: Es un documento que se origina dentro del contexto académico y científico. Tiene como finalidad difundir el conocimiento generado de un trabajo de investigación, ya sea experimental o de reflexión y análisis, a la comunidad científica y la sociedad en general (Padrón Guillén, 1996).

Referencia Bibliográfica: Uno de los componentes más importantes de un texto académico son las referencias a otras fuentes de información, las cuales el autor o los autores emplean para sustentar sus contenidos.

Rodrigues Alves y cols. (2018) definen Referencia bibliográfica como una secuencia de texto contiguo que contiene la información necesaria para poder encontrar y consultar todo acerca de una fuente de información.

Una referencia suele componerse de diversos componentes, como autor(es), título, año y editorial, entre otros. Los cuales se codifican según el estilo de citación seleccionado y el tipo de publicación al que hacen referencia. Por lo general, las referencias se concentran en una sección específica dentro de un documento académico (ver Figura 2.1).

Es importante mencionar que las referencias bibliográficas son la base del análisis de citas el cual es un tipo de estudio bibliométrico. Dicho análisis permite evaluar la actividad científica de una persona, de un grupo, una institución, un país o una región (López Piñero, 1972).

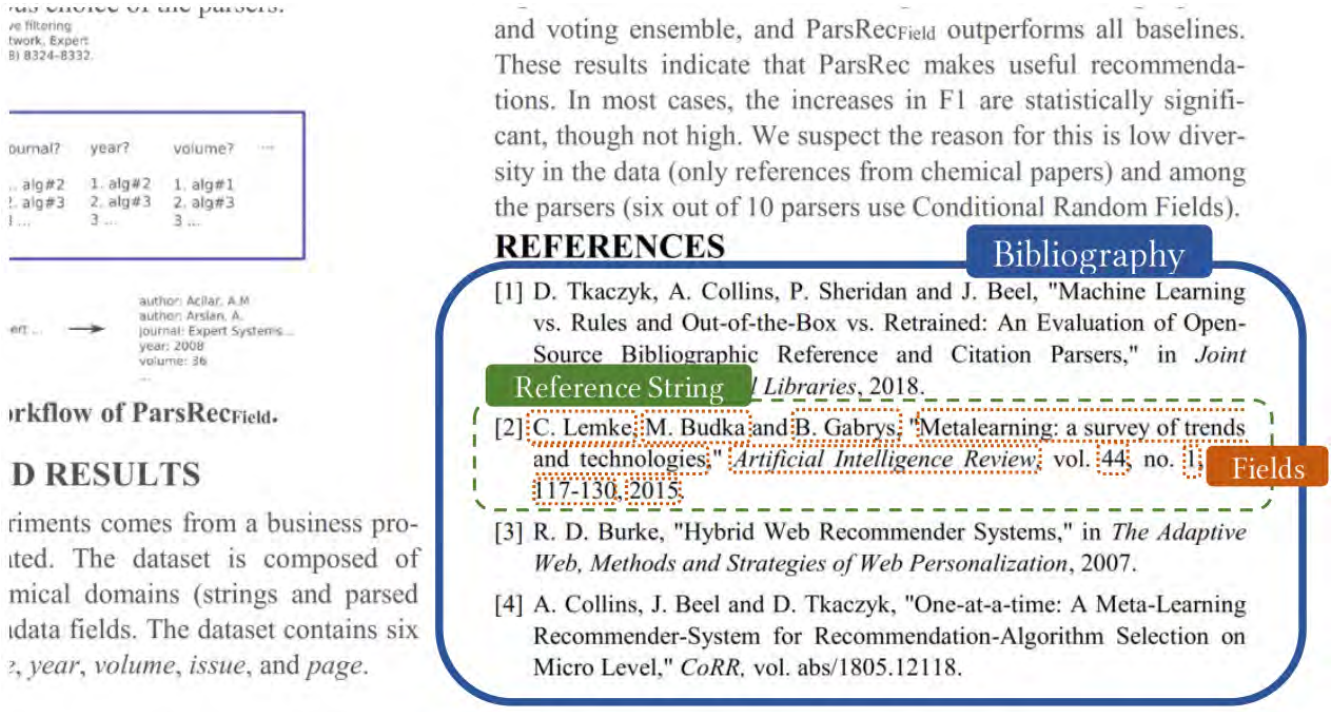


Figura 2.1: Ejemplo de Referencia Bibliográfica (Grennan y cols., 2019).

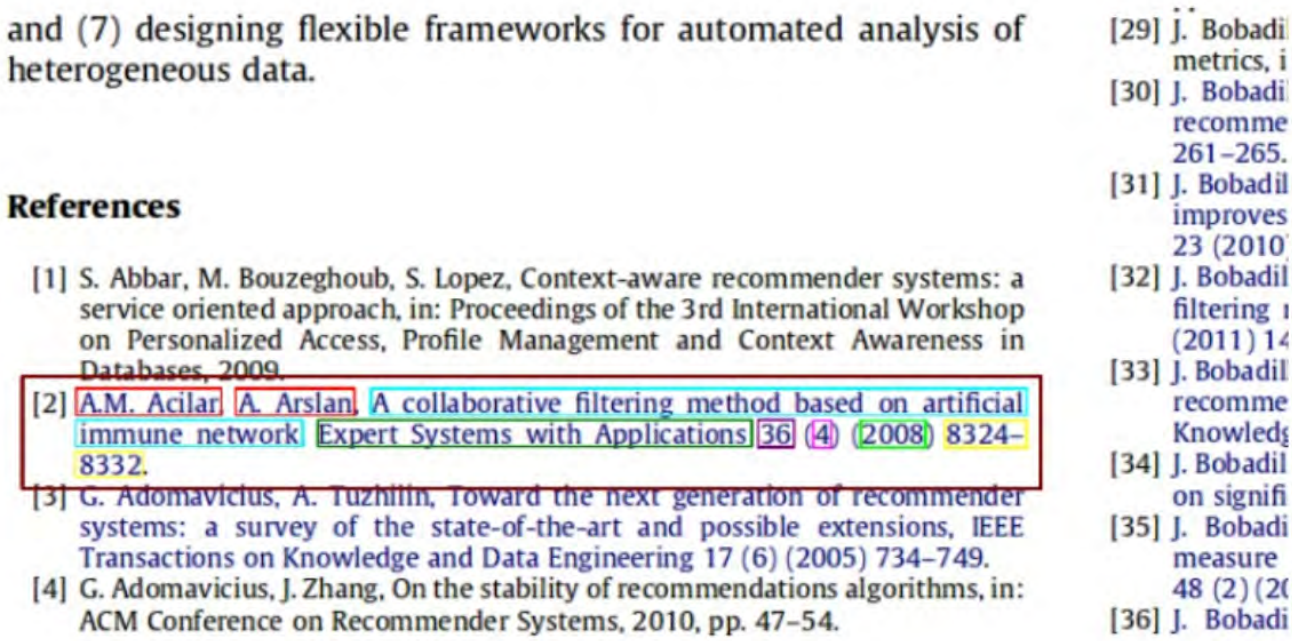
2.1.2. Minería de referencias

Rodrigues Alves y cols. (2018) definen Minería de referencias (RM por sus siglas en inglés) como la tarea de analizar sintácticamente referencias bibliográficas, mediante técnicas de procesamiento del lenguaje natural. Con la finalidad de resolver las subtareas de detección, extracción y segmentación.

Las subtareas antes mencionadas se describen de la siguiente manera:

- Detección: Tarea que consiste en detectar una zona dentro del texto que contenga una o más referencias bibliográficas (de ahora en adelante llamadas cadenas de referencia).
- Extracción: Tarea que consiste en separar y extraer cada una de las cadenas de referencia detectadas en la tarea anterior.
- Segmentación: Tarea que consiste en analizar cada cadena de referencia y descomponerla en sus componentes para transformarlos en registros de metadatos.

En la Figura 2.2 se muestra una representación de las tres tareas descritas anteriormente. Se identifica la zona donde se encuentran las cadenas de referencia, en particular la sección "**References**" (correspondiente a Detección). Luego, se extrae una cadena de la sección, como la que tiene el identificador "**[2]**" (relacionada con la Extracción). Finalmente, se segmentan los componentes, lo que se refleja en los cuadros de colores (correspondiente a la Segmentación).



and (7) designing flexible frameworks for automated analysis of heterogeneous data.

References

[1] S. Abbar, M. Bouzeghoub, S. Lopez, Context-aware recommender systems: a service oriented approach, in: Proceedings of the 3rd International Workshop on Personalized Access, Profile Management and Context Awareness in Databases, 2009.

[2] A.M. Acilar, A. Arslan, A collaborative filtering method based on artificial immune network, Expert Systems with Applications 36 (4) (2008) 8324–8332.

[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.

[4] G. Adomavicius, J. Zhang, On the stability of recommendations algorithms, in: ACM Conference on Recommender Systems, 2010, pp. 47–54.

[29] J. Bobadi metrics, i

[30] J. Bobadi recomme 261–265.

[31] J. Bobadil improves 23 (2010

[32] J. Bobadil filtering (2011) 14

[33] J. Bobadil recomme Knowledg

[34] J. Bobadil on signifi

[35] J. Bobadi measure 48 (2) (2(

[36] J. Bobadi

Figura 2.2: Representación de las tres subtareas de la Minería de referencias (Tkaczyk, Collins, y cols., 2018).

En la Figura 2.3 se muestra un registro de metadatos con los componentes segmentados de la cadena de referencia analizada en el ejemplo de la Figura 2.2.

```
author:  Acilar, A.M.
author:  Arslan, A.
title:   A collaborative filtering method based on artificial immune network
journal: Expert Systems with Applications
volume:  36
issue:   4
year:    2008
pages:   8324-8332
```

Figura 2.3: Ejemplo de un Registro de metadatos, resultado de la Minería de referencias (Tkaczyk, Collins, y cols., 2018)

2.1.3. Procesamiento del lenguaje natural

El procesamiento del lenguaje natural (NLP por sus siglas en inglés) es una disciplina que surge de la tríada entre computación, lingüística e inteligencia artificial. Tiene como propósito estudiar y flexibilizar las interacciones humano-máquina.

> *"Se ocupa de la formulación e investigación de mecanismos computacionales para la comunicación entre personas y máquinas por medio del lenguaje natural, es decir, de las lenguas del mundo. No trata de la comunicación por medio de lenguas naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse que sean eficaces computacionalmente."*
>
> Wikipedia (2021b)

Con tal finalidad, las técnicas de NLP pueden clasificarse en tres categorías (Galicia y cols., 1998):

- Reconocimiento de patrones.
- Correlaciones estadísticas.
- Análisis pragmático.

Las subtareas que componen la RM suelen considerarse como problemas de etiquetado de secuencias. Por ello, varias de las técnicas de NLP que se han usado para resolverlas, corresponden a la categoría de etiquetado de secuencias (Councill y cols., 2008; Prasad y cols., 2018).

2.1.4. Etiquetado de secuencias

El etiquetado de secuencias es considerada una tarea que se resuelve mediante algoritmos de aprendizaje automático (ML, por sus siglas en inglés) supervisado. Consiste en asignar una etiqueta a cada elemento de una secuencia de lenguaje dada (voz o texto principalmente).

Los usos más comunes de tales algoritmos son (Rodrigues y cols., 2014):

- Etiquetado de partes de una oración.
- Segmentación.
- Identificación de entidades nombradas.
- Predicción genética.

En el caso de la RM, se busca identificar las palabras dentro de un texto académico que correspondan a los componentes de una referencia y asignarles una etiqueta. Esto se observa con mayor claridad en la Figura 2.4, que muestra la asignación de etiquetas a cada componente de una referencia bibliográfica procesada por el sistema SciWing (Kashyap y Kan, 2020), representando la subtarea de segmentación de la RM.

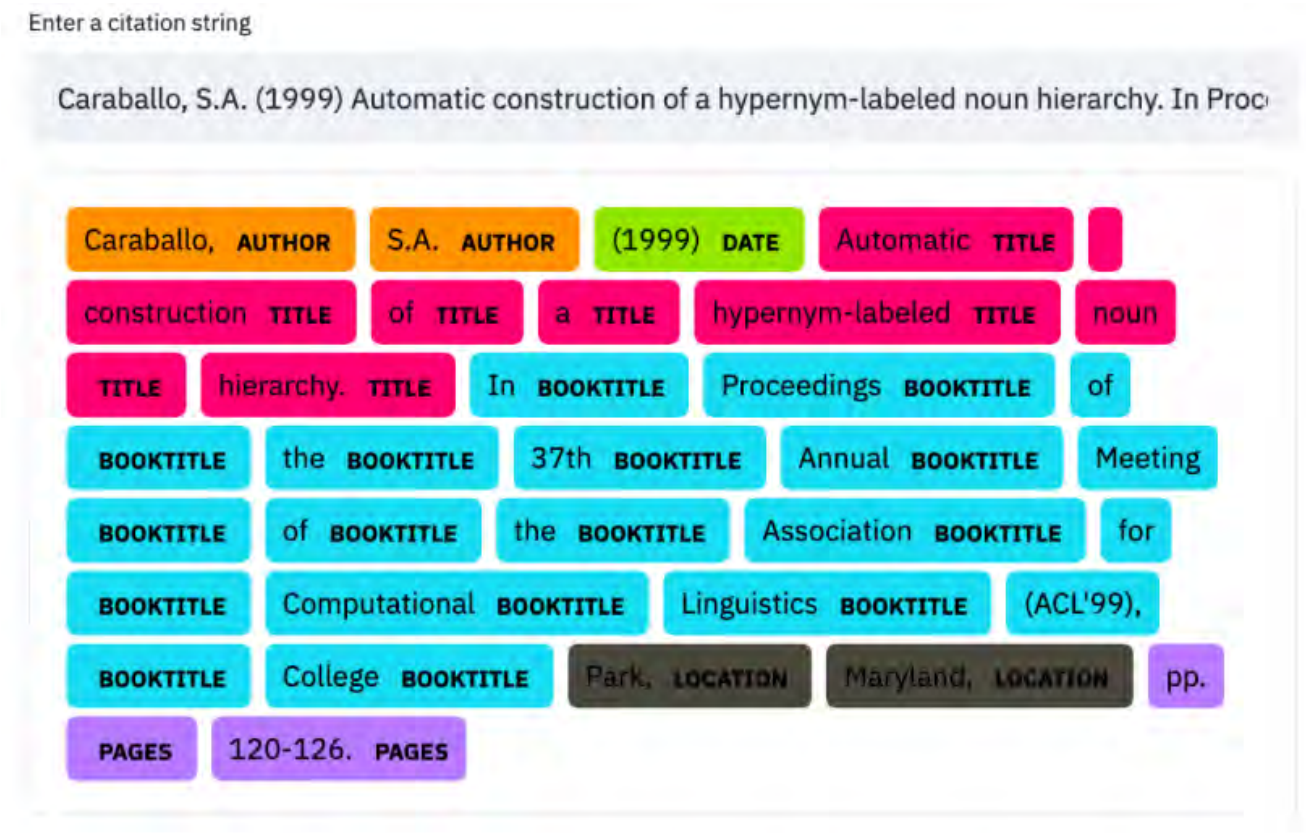


Figura 2.4: Ejemplo de una referencia etiquetada utilizando SciWing (Kashyap y Kan, 2020)

2.1.5. Aprendizaje automático

El aprendizaje automático es una disciplina que surge de la intersección entre las Ciencias de la Computación y la Inteligencia Artificial. Su objetivo consiste en desarrollar técnicas y métodos que les otorguen a los algoritmos la capacidad de aprender. En otras palabras, las computadoras mejoran su desempeño en una tarea mediante la experiencia (Russell y Norvig, 2004).

> *"En el aprendizaje de máquinas, un computador observa datos, construye un modelo basado en esos datos y utiliza ese modelo a la vez como una hipótesis acerca del mundo y una pieza de software que puede resolver problemas."*
>
> Wikipedia (2021a)

Se considera que un algoritmo ha "aprendido" cuando puede realizar inferencias acertadas, basándose en la exposición que haya tenido a conjuntos de datos para su entrenamiento. En palabras de Chollet (2017), el algoritmo transforma los datos de entrenamiento en representaciones útiles que se aproximan a la salida esperada. De manera que logra descubrir las reglas que permiten automatizar la tarea que se pretende resolver.

Aprendizaje supervisado

El aprendizaje supervisado es una rama del aprendizaje automático que se emplea en la resolución de problemas de etiquetado de secuencias, es decir, un problema de clasificación. El objetivo es clasificar vectores de datos mediante la asignación de una etiqueta.

Para lograrlo, el algoritmo debe ser expuesto a un conjunto suficiente de vectores ya etiquetados previamente. De manera que, el algoritmo puede ajustar sus parámetros internos, y así ser lo más acertado posible al asignar etiquetas a vectores a los que nunca ha sido expuesto. En otras palabras, el algoritmo únicamente necesita experimentar en una pequeña parte ya clasificada de un universo de datos para aprender a clasificar los restantes.

Dentro del contexto del RM el enfoque más usado es el probabilístico. Su finalidad es calcular la probabilidad de que una palabra (token), corresponda a una etiqueta (clase) dada, teniendo en cuenta observaciones anteriores.

2.1.6. Aprendizaje profundo

El aprendizaje profundo (DL por sus siglas en inglés) es un subdominio del aprendizaje automático. El DL comprende algoritmos basados en redes neuronales artificiales (ANN en inglés) organizadas en un número variable de capas. Con el propósito de aprender representaciones de datos que cuentan con múltiples niveles de abstracción.

El término "profundo" hace referencia al número de capas (LeCun y cols., 2015). Los algoritmos basados en DL se distinguen de otros algoritmos de aprendizaje automático gracias a una serie de características descritas por Chollet (2017):

- Robustez: permite filtrar el ruido natural de los datos y, por lo tanto, ser tolerante a problemas de desempeño.
- Generalización: permite la reutilización de un mismo algoritmo para otras aplicaciones y/o tipos de datos.
- Escalabilidad: el rendimiento suele tener un margen de mejora cuando se amplían la cantidad de datos y se mejora la calidad de los mismos, generando una mayor asertividad a las inferencias hechas por el algoritmo.

Agrawal y cols. (2019) describe a las ANN, como mecanismos computacionales que se inspiran en los mecanismos neuronales de los seres vivos, es decir, emulan el funcionamiento de las neuronas biológicas y sus conexiones. Las ANN emplean múltiples unidades computacionales que establecen conexiones entre ellas, que están reguladas por una medida conocida como peso o tensor, lo cual es análogo a la conexión sináptica en las estructuras neuronales biológicas.

Una forma de entender el concepto de ANN es visualizarlas como un grafo computacional. Dicho grafo se compone de una serie de funciones simples y recursivas que tienen como propósito aprender funciones más complejas. Una ANN se compone de una capa de un número variable de neuronas de entrada, una o múltiples capas de un número variable cada una de neuronas, denominada en su conjunto, como capa oculta, y una capa con una o más neuronas que representa la salida.

En la Figura 2.5 se puede apreciar un ejemplo de una red neuronal con tres neuronas de entrada, una capa oculta compuesta por cuatro neuronas y su salida corresponde al resultado de una tarea de clasificación.

En el contexto del RM, los tipos de redes de ANN más empleadas para resolver las subtareas corresponden a dos tipos:

- Redes neuronales convolucionales se emplean para **detección** y **extracción**.
- redes neuronales recurrentes para **detección**, **extracción** y **segmentación**.

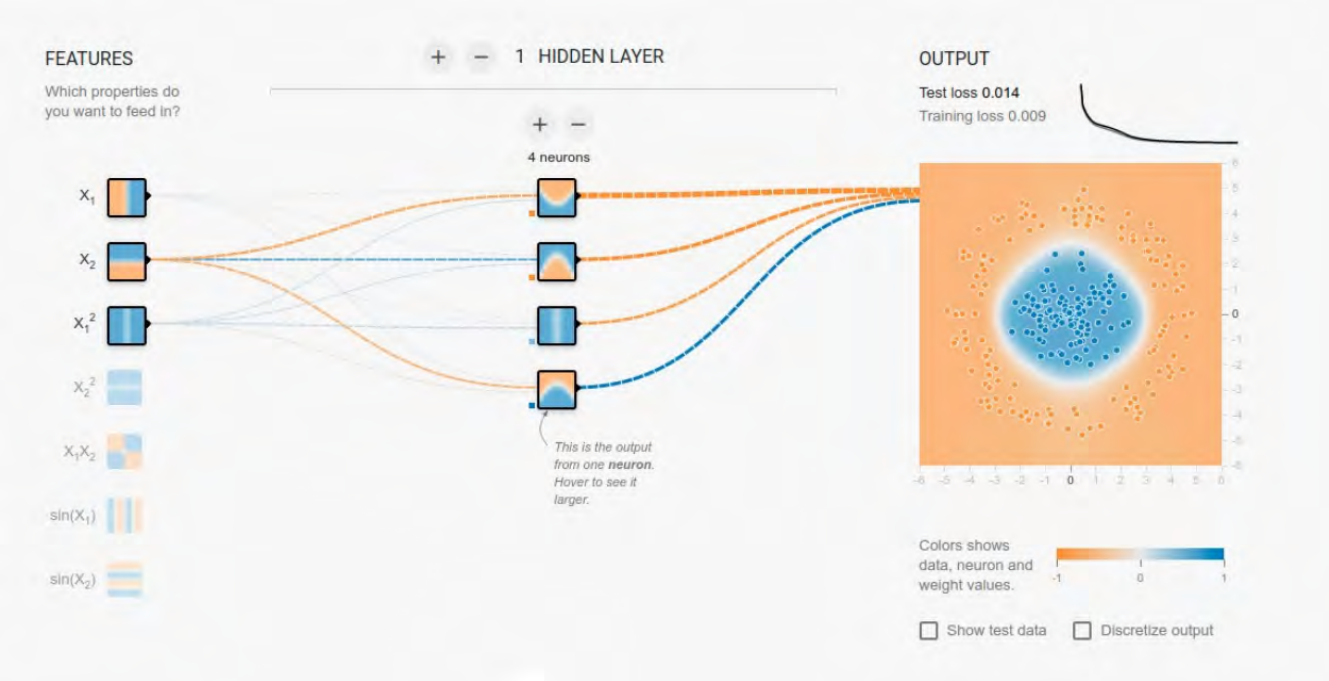


Figura 2.5: Ejemplo de una red neuronal empleando un simulador. Fuente: `https://playground.tensorflow.org/`

Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN por sus siglas en inglés), son redes multicapa (a mayor cantidad de capas la red se considera más profunda), que suelen emplearse en tareas de visión computacional.

Las CNN aprenden de manera incremental, es decir, la información de entrada va pasando capa por capa, atravesando en cada una por una serie de transformaciones que van identificando jerarquías de características, a mayor profundidad de la red, mayores características de una imagen es capaz de abstraer (Chollet, 2017), en la Figura 2.6 se puede apreciar un ejemplo de las capas de un modelo CNN y cómo extraen dichas características.

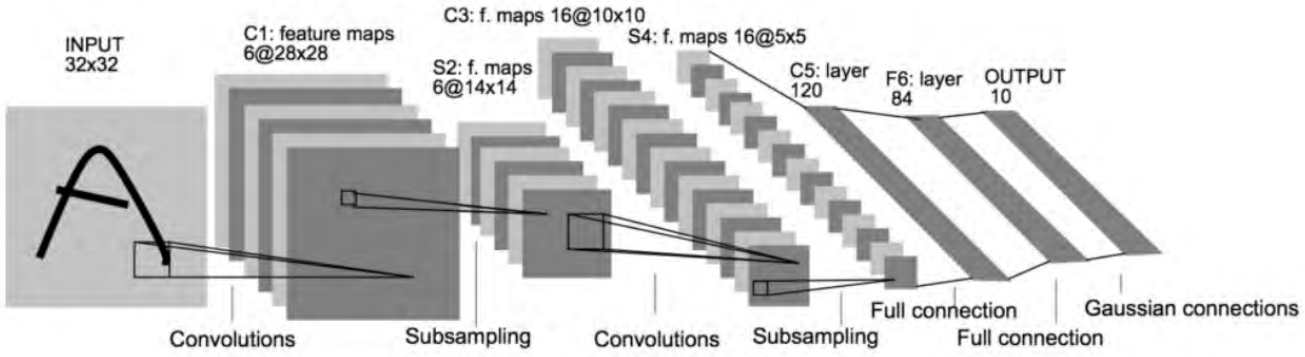


Figura 2.6: Ejemplo de funcionamiento de una Red Neuronal Convolucional. Recuperada de: `https://sitiobigdata.com/2019/05/01/innovaciones-arquitectonicas-redes-neuronales-clasificacion-imagenes/`

En el contexto del RM, las CNN se han empleado para las subtareas de detección y extracción (Bhardwaj y cols., 2017), se puede apreciar un ejemplo en la Figura 2.7, donde se emplea una imagen de una hoja de referencias bibliográficas.

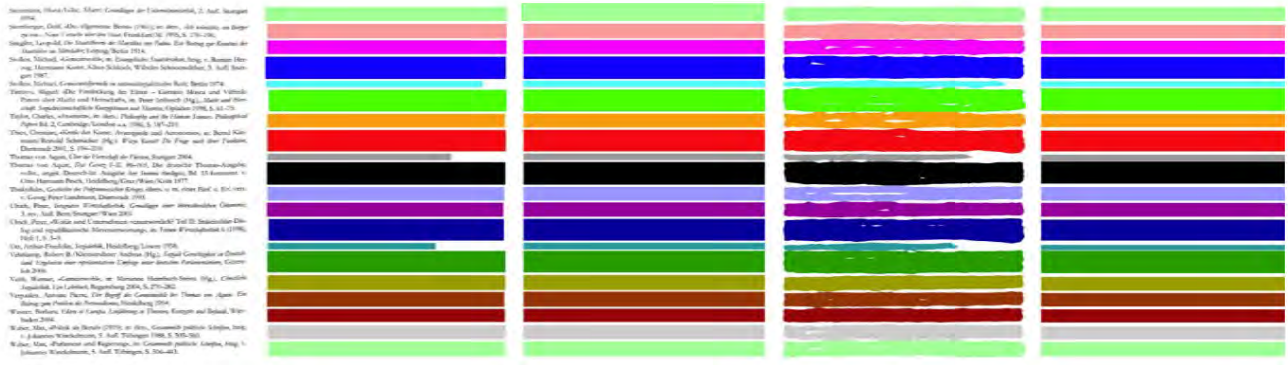


Figura 2.7: Ejemplo de la entrada y salida de una CNN aplicada a las subtareas de detección y extracción de la Minería de referencias. (Bhardwaj y cols., 2017)

Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN por sus siglas en inglés) son redes que se especializan en analizar datos secuenciales. Utilizan la recurrencia como "memoria" mediante un elemento de estado oculto que se actualiza en cada iteración. En otras palabras, pueden "recordar" la información previa que han analizado para tenerla como factor de cálculo al momento de procesar nueva información (Ketkar, 2017).

En palabras de Gupta y Raza (2019), la salida de una RNN se utiliza como entrada para el siguiente paso en la secuencia, esto se puede apreciar en la Figura 2.8. Las RNN se emplean principalmente para el procesamiento de información dinámica, como son las series de tiempo y el control de procesamiento.

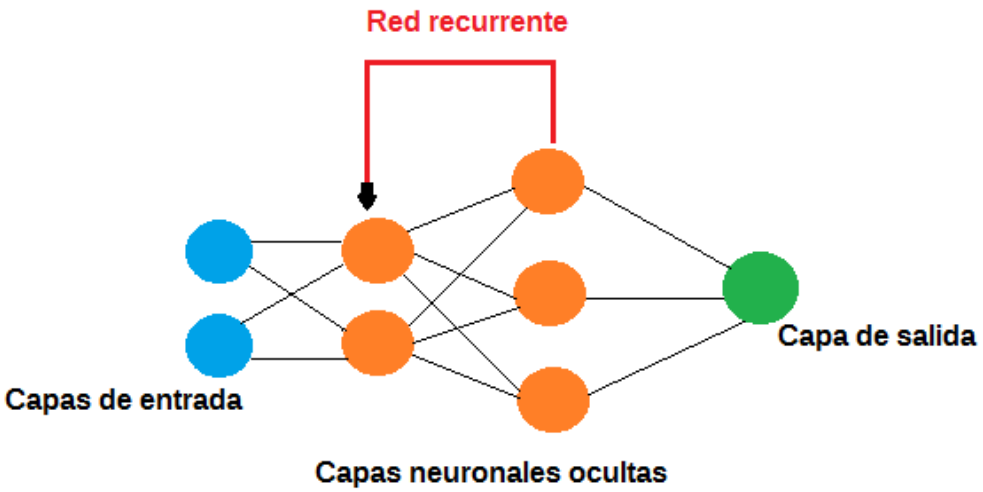


Figura 2.8: Ejemplo de una Red Neuronal Recurrente con una neurona de la capa oculta retroalimentada por la salida posterior. Fuente: `https://inteligencia-artificial.dev/tipos-redes-neuronales/`

En el contexto del RM, existe un subtipo de las RNN, llamadas Redes Neuronales con Memoria a Corto y Largo Plazo (LSTM en inglés) que se suelen emplear para las subtareas de detección, extracción y segmentación (Rodrigues Alves y cols., 2018).

Las RNN son ideales para las subtareas de RM dado que no adolecen de las limitaciones de las ANN clásicas, que es el hecho de que los estados ocultos solamente se mantienen por una iteración. En el caso de las redes LSTM, los estados ocultos se pueden mantener más allá de un determinado número de iteraciones y cambian con base en la relevancia de la salida de nuevas iteraciones (Prasad y cols., 2018).

Transformadores

Los Transformadores proponen una arquitectura de ANN, que se compone de un codificador y un decodificador (ver Figura 2.9), para el análisis de secuencias. Eliminan el uso de convoluciones y recurrencias, apoyándose únicamente en mecanismos de atención, tales cualidades permiten extraer características como sintaxis e incrustaciones (embeddings en inglés), sin requerir grandes volúmenes de datos etiquetados.

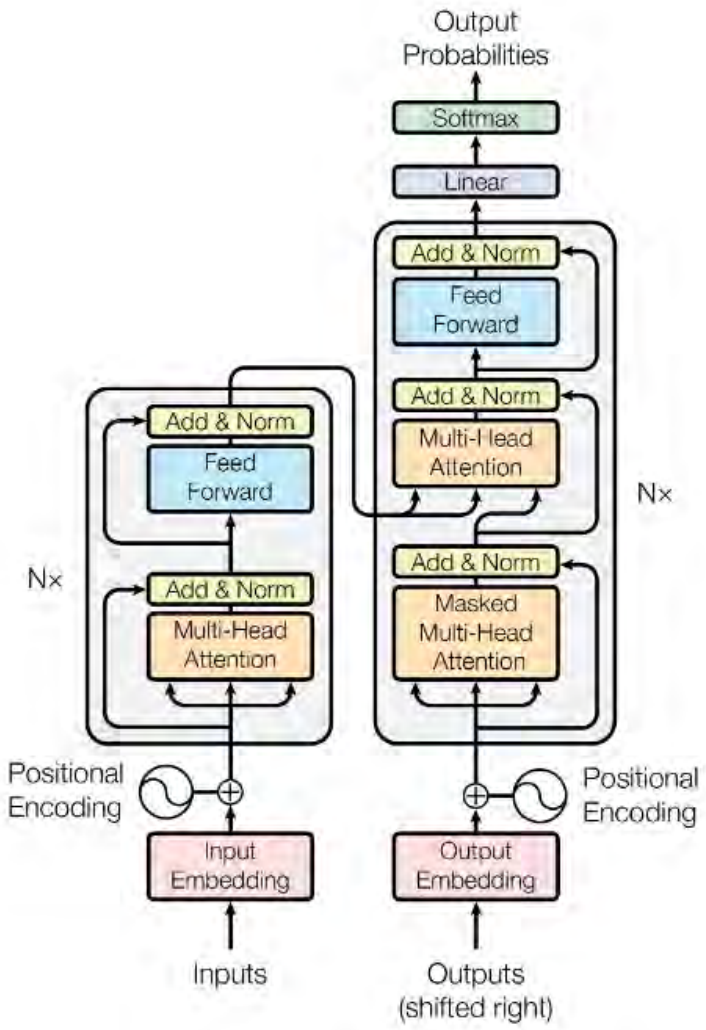


Figura 2.9: Arquitectura de un modelo basado en Transformadores (Vaswani y cols., 2017)

Adicionalmente, los Transformadores han mostrado una mayor eficiencia en tareas de traducción automática, al tiempo que su entrenamiento es paralelizable y requiere menor tiempo que las RNN o las CNN (Vaswani y cols., 2017).

2.2. Estado del Arte

A lo largo de la sección se describen, de manera breve, las publicaciones más relevantes relacionadas con el abordaje de los subproblemas (descritas en la sección 2.1.2) que componen la Minería de referencias. El trabajo previo en RM puede clasificarse en dos diferentes grupos que a su vez pueden tener diversas categorías y niveles dependiendo de sus aproximaciones.

Estos grupos se pueden apreciar en la Figura 2.10, a continuación, se describirá cada grupo con respecto a sus enfoques, técnicas y las subtareas que resuelven.

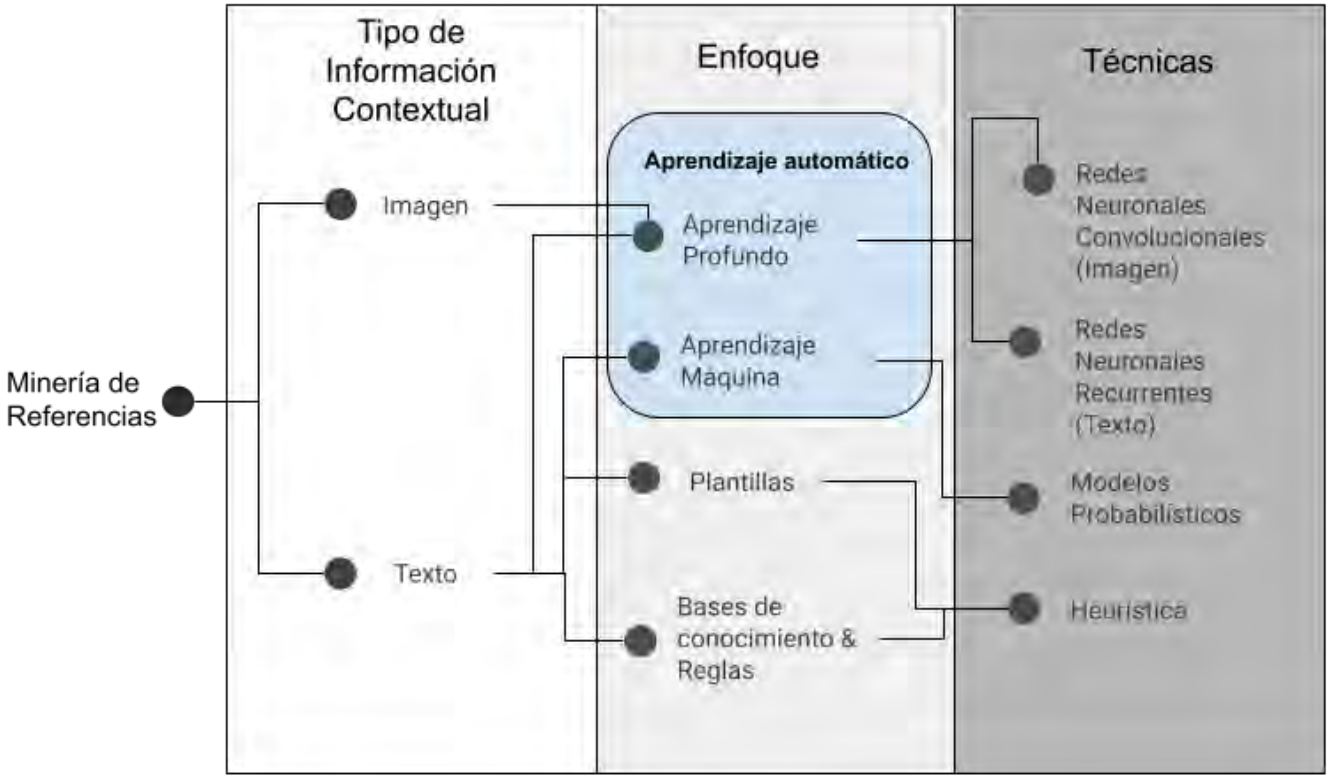


Figura 2.10: Agrupaciones de los distintos abordajes de Minería de referencias y sus niveles.

2.2.1. Imagen como información contextual

En este grupo se encuentran las propuestas que emplean imágenes como base de información contextual (Rizvi y cols., 2019; Bhardwaj y cols., 2017). Las propuestas se inspiran en la visión humana, es decir se enfocan en el problema utilizando visión computacional, empleando CNN, para detectar las cadenas de referencias en la imagen de un documento de texto. Las ventajas son las siguientes:

- Son ideales (en palabras de los autores) para trabajar documentos que no han nacido digitalmente.
- Evitan los errores del proceso de Reconocimiento óptico de caracteres (OCR por sus siglas en inglés).
- Son independientes del idioma.

Como desventaja, los enfoques que emplean imágenes como información contextual, solamente son útiles para cubrir las subtareas de detección y extracción de cadenas de referencias. La propuesta de Rizvi y cols. (2019) subsana tal desventaja aplicando OCR a las cadenas de referencias extraídas y empleando campos aleatorios condicionales para la tarea de segmentación

Cabe resaltar que Bhardwaj y cols. (2017) empleó una implementación de una Red Neuronal Totalmente Convolucional, también conocida como FCN-8, dicha implementación lleva el nombre de DEEPBibx. Mientras que Rizvi y cols. (2019) utiliza una CNN con máscaras de región (Mask R-CNN) y su implementación lleva el nombre de DeepBird.

2.2.2. Texto como información contextual

Grupo donde se concentran la mayor cantidad de propuestas, todas ellas presentan diversos enfoques que pueden ser categorizados de la siguiente manera.

Enfoques basados en aprendizaje máquina

En este grupo se encuentran las propuestas que se basan en modelos probabilísticos, los cuales aprenden a partir de conjuntos etiquetados de referencias bibliográficas, es decir con texto, y suelen cubrir todas las subtareas de la Minería de referencias.

Algunos de los trabajos en RM de dicha categoría, abordan la subtarea de detección mediante la identificación de la sección de referencias en un documento, para lo cual emplean máquinas de sSoporte vectorial (SVM por sus siglas en inglés)(Zou y cols., 2010; Tkaczyk y cols., 2015).

Mientras que otros (Lopez, 2009; Körner, 2017) se decantan por el uso Campos Aleatorios Condicionales (CRF por sus siglas en inglés), debido a su capacidad de modelar límites al momento de decidir entre diferentes clases para asignar etiqueta a una palabra (token).

En cuanto a las subtareas de extracción y segmentación, se han empleado principalmente SVM, CRF y Modelos Ocultos de Markov (HMM, por sus siglas en inglés). Dado que dichas subtareas corresponden al problema de etiquetado de secuencias, HMM parece ser la herramienta adecuada para estimar la etiqueta de un token con base en estados detectados anteriormente (Boukhers y cols., 2019).

Hetzner (2008) aplica un modelo basado en HMM simple de primer orden, logrando aproximarse al 90 % en un conjunto de datos de referencias homogéneas del dominio de ciencias de la salud. En tanto que Yin y cols. (2004) logra afinar los resultados empleando una variante del HMM, usando Bigramas, que ajusta la probabilidad de estimación sin modificar la estructura propia de un HMM.

Por su parte, Ojokoh y cols. (2011) empleó otra variante de HHM, usando Trigramas. Tiene como particularidad un tamaño de ventana de contexto más amplio al momento de estimar la probabilidad de que un token corresponda a una etiqueta determinada.

En 2011 Zhang y cols., propusieron una SVM estructural para segmentar referencias biomédicas, logrando una precisión del 98 %, atribuida a la regularidad estructural del estilo. Zou y cols. (2010) comparó SVM y CRF, encontrando precisiones similares (97 % aproximadamente).

Peng y McCallum (2006a) fueron los primeros en aprovechar los modelos CRF mediante variaciones gaussianas. Estos modelos, una especialización de los HMM, influyen en el cálculo de la probabilidad de que un token corresponda a una etiqueta específica, considerando las etiquetas de iteraciones anteriores.

Romanello y cols. (2009) combinó CRF con un analizador de N-Grams (subsecuencia de **n** elementos) para procesar un tipo especial de referencia, conocido como “Canónicas”, que corresponde a textos clásicos.

Los modelos CRF se han convertido en un referente en la Minería de referencias, al punto de ser empleados en las primeras herramientas públicas, como GROBID (Lopez, 2009) y ParsCit (Councill y cols., 2008). Tkaczyk y cols. (2015) publican su implementación CERMINE, la cual resuelve el flujo completo de todas las subtareas de la RM, empleando enfoques heurísticos y de ML, como son:

- SVM

- K-Means
- CRF

Logrando buenos resultados y permitiendo que personas no expertas puedan hacer RM.

A diferencia de los demás enfoques, Körner (2017) propone un modelo basado en CRF que en lugar de etiquetar palabras, toma en consideración todas las líneas de un documento, en lugar de solamente identificar la sección de referencias. Así, el modelo clasifica líneas en lugar de palabras individuales, disminuyendo la complejidad del modelo a utilizar y permite extraer cadenas de referencias contenidas en todo el documento y no solamente en una sección.

Se debe resaltar que, el enfoque antes mencionado es únicamente útil para las tareas de identificación y extracción de la RM. Resulta relevante dado que implica un menor número de variables aleatorias, lo que resulta eficiente cuando se considera un entrenamiento con una cantidad relativamente baja de datos etiquetados manualmente; en una posterior publicación Körner y cols. (2017) comparan su propuesta con otras, como GROBID, empleando el idioma alemán.

Boukhers y cols. (2019), presentan una propuesta holística para abordar el problema de la Minería de referencias. Diseñaron un flujo que contempla las subtareas de detección, extracción y segmentación de la RM (a diferencia de otras propuestas que solamente se concentran en una o dos subtareas), con la finalidad de tener un esquema coherente que reduce los errores y sigue una aproximación probabilística.

La propuesta funciona en dos fases correlacionadas. La primera es la clasificación de líneas, la cual mediante un modelo entrenado basado en Bosques Aleatorios (RF, por sus siglas en inglés) se etiqueta cada línea de un texto que pertenece a una referencia, clasificándolas en tres tipos: primera línea, línea intermedia y última línea.

De manera que se extrae cada cadena de referencia para posteriormente entrar en la segunda fase, que consiste en segmentar las cadenas de referencia extraídas mediante campos aleatorios condicionales. Concluyendo que se supera a otros enfoques, dada su aproximación holística.

Adicionalmente, se resalta el hecho de que este enfoque puede ubicar referencias que no están contenidas en una sección específica sino distribuidas en todo el documento, común en ciencias sociales y humanidades.

Es importante mencionar un comentario hecho por Yang y cols. (2020) sobre los enfoques basados en aprendizaje máquina, quienes señalan que su precisión depende de características específicas y generadas manualmente, por lo tanto, son dependientes del dominio que se intenta estudiar.

Es decir, los modelos entrenados en un dominio no suelen ser aptos para trasladarlos tal cual a otro dominio y/o estilo de referencia bibliográfica, lo que implica la necesidad de generar datos etiquetados, reentrenar los clasificadores y probablemente ajustar sus parámetros.

Enfoques basados en aprendizaje profundo

Boukhers y cols. (2019), consideran que la deficiencia de la poca generalización de los datos, en los enfoques basados en ML, puede ser superada empleando Redes Neuronales profundas, dado que pueden generalizar de una forma más precisa los datos con los cuales se entrenan.

La primera propuesta basada en redes neuronales para la minería referencias fue hecha por Prasad y cols. (2018), quienes diseñaron una Red Neuronal de tipo LSTM para obtener representaciones de

los tokens que corresponden a una referencia y, con las características extraídas, entrenar un modelo basado en CRF.

De modo que se creó un modelo sólido con características extraídas por la RNN, evitando así la necesidad de extraer manualmente las características de entrenamiento; dando origen a la implementación *Neural Parscit*.

El último de los trabajos en esta línea es el de Rodrigues Alves y cols. (2018), quienes emplearon una arquitectura Bi-LSTM-CRF, donde emplean capas de predicción SoftMax-CRF e incrustaciones de palabras, basadas en Word2Vec Mikolov y cols. (2013), y caracteres (Ling y cols., 2015). Con la finalidad de poder identificar referencias en todo un documento de texto[1] y no solamente en la sección referencias.

Es importante tener en cuenta lo señalado por Grennan y Beel (2020), quienes identifican un sesgo significativo en los enfoques basados en aprendizaje profundo: la insuficiente cantidad de datos etiquetados para un entrenamiento eficaz. A manera de solución, compararon la eficiencia de modelos entrenados con datos reales y etiquetados manualmente con lo que llaman "cadenas de referencia sintéticas".

Las cadenas sintéticas consisten en referencias reales en formato Bibtex, convertidas a cadenas de referencia mediante un script. Los autores concluyen que los datos sintéticos son muy adecuados para el entrenamiento de modelos destinados al análisis de citas.

Uno de los primeros en utilizar transformadores fue Uddin (2022), quien presenta un enfoque basado en un codificador de transformador para la subtarea de Segmentación, el cual fue entrenado con un conjunto de datos de más de 220k referencias y en palabras del autor logra el 94.2 % de F1.

Por último se debe mencionar a Choi y cols. (2023), quien propone un enfoque basado en el modelo preentrenado Bert-base-multilingual-cased, basado en la tecnología Bidirectional Encoder Representations from Transformers (Devlin y cols., 2018), el cual logra el 99.83 % de F1.

Enfoques basados en plantillas

Estos enfoques resultan eficientes para la subtarea de segmentación de la RM debido a su capacidad para representar de manera general los patrones. Las plantillas se diseñan con base al conocimiento previo del dominio, por lo que extraen información relevante de textos cuando las condiciones definidas en las mismas coinciden Boukhers y cols. (2019).

A pesar de su eficiencia, el diseño de las plantillas requiere de una considerable cantidad de esfuerzo humano, y dada la naturaleza no uniforme (en la práctica) de la escritura de referencias, no se suelen considerar un enfoque factible.

Chowdhury (1999) realiza el primer intento de resolverlo, proponen una plantilla de formato para la escritura de la referencia por parte de los autores, con el fin de reducir las citas irregulares y, por lo tanto, la complejidad en el diseño de las plantillas.

Por su parte Ding y cols. (1999) emplearon un mayor número de plantillas (tres en lugar de una) para solventar el problema de patrones irregulares en la escritura de las referencias, logrando buenos resultados para extraer componentes con poca variabilidad, como son autores o nombres de revistas, en cambio, no tuvo los mismos resultados para otros componentes.

[1] Dominios como el de Humanidades y Ciencias Sociales suelen contener referencias en distintos lugares del documento.

También en 2007, Day y cols. proponen INFOMAP otra implementación que emplea plantillas jerárquicas, las cuales les permiten superar la falta de personalización de las reglas mediante una estructura de árbol jerárquico. Así es capaz de representar diversos patrones para los componentes de diferentes estilos de referencias.

Por último en 2008, Chen y cols. proponen BibPro, una implementación que mejoró el rendimiento de trabajos anteriores reemplazando la base de conocimiento por símbolos de puntuación para identificar el formato de referencia, reportando mejores resultados que (Councill y cols., 2008) empleando un conjunto de datos de prueba que contenía seis estilos de referencias diferentes.

Enfoques basados en reglas y bases de conocimiento

Los enfoques basados en reglas y bases de conocimiento se sustentan en el diseño de reglas mediante heurística y bases de conocimiento para poder afrontar las tareas de extracción y segmentación de la RM, los principales exponentes son: Cortez y cols. (2007) con su propuesta llamada Flux-CiM y Hsieh y cols. (2014) con FBA.

Cortez y cols. (2007) presentan Flux-CiM, el cual emplea una base de conocimientos (KB, por sus siglas en inglés) para la extracción de referencias, el enfoque hace mucho énfasis en que no requiere de un proceso de entrenamiento, como es el caso de los modelos basados en aprendizaje automático y/o profundo, logrando gran flexibilidad y sostenibilidad; toda su potencia proviene de un KB que consiste básicamente en una colección de referencias estructurada.

El proceso consiste en identificar cada cadena de referencia y segmentar las palabras en bloques (análogo al concepto de tokenizar) e identificar los marcadores de separación (los signos de puntuación como comas, puntos dobles, puntos etc.) y estimar la probabilidad de que un valor aparezca en cierto segmento tomando en cuenta lo que existe en el KB y los valores encontrados en determinado campo, tomando en cuenta a los bloques vecinos para incrementar la precisión.

Los resultados muestran un 93 % de efectividad, en sus pruebas, donde se aplicó a dos dominios diferentes (ciencias computacionales y ciencias de la salud).

Por su parte, Hsieh y cols. (2014) presentan un enfoque basado en marcos (FBA por sus siglas en inglés), el cual propone construir una representación flexible de la información, la cual es construida por expertos que compilan manualmente abreviaturas, patrones límite de posiciones, y prefijos; para finalmente emplear un algoritmo de aproximación de coincidencia y así reconocer los componentes deseados de las cadenas de referencias.

Finalmente, se comparó con una solución basada en CRF en dominios diferentes, logrando una reducción del error cercana al 70 %.

Otros enfoques

Vale la pena mencionar que recientemente se han presentado propuestas que no corresponden con ninguna de las categorías anteriores, el primer caso es el de Tkaczyk, Gupta, y cols. (2018), con la propuesta llamada ParsRec, que tiene la peculiaridad de ser un enfoque basado en la recomendación y meta-aprendizaje.

Su principal premisa es “A pesar de que existen multitud de analizadores de referencias, que abordan el problema desde diferentes enfoques, ninguno de ellos ofrece óptimos resultados en todos los escenarios”.

ParsRec recomienda el analizador más adecuado, de diez contemplados, con base en la cadena de referencia en turno. Los resultados son considerados prometedores por parte de los autores, dado que muestran el potencial del sistema. La combinación de ambos enfoques propuestos supera al mejor analizador hasta un 18.9 % en la reducción de tasa de error.

El último caso es el de Yang y cols. (2020) quienes proponen el Enfoque basado en Principios (PBA por sus siglas en inglés) para abordar la subtarea de extracción y segmentación de la RM.

Consiste en un generador automático de plantillas que captura patrones mediante un algoritmo de conjunto dominante, tales plantillas son capaces de extraer referencias empleando una técnica de coincidencia de plantillas, basada en alineación, que utiliza un modelo de regresión logística, lo que lo hace más general y flexible que los enfoques basados en reglas.

Esta propuesta ha sido comparada con otros enfoques existentes, mostrando un mayor rendimiento en diversos corpus. En palabras de los autores, su enfoque toma lo mejor de los enfoques basados en plantillas y el modelado estadístico.

Se concluye que la principal contribución del PBA es mantener la explicabilidad de los métodos basados en plantillas mientras se aprovecha la optimización que aportan los modelos estadísticos de aprendizaje automático.

Comparativa

Para finalizar la sección de enfoque, es importante mencionar que Tkaczyk, Collins, y cols. (2018) realizaron una comparación entre diferentes enfoques para la extracción de citas, como son: Campos aleatorios condicionales (CRF), Expresiones regulares, reglas, comparación de plantillas y Redes neuronales LSTM.

A pesar de que se realiza una breve descripción del estado del arte, se aclara que no fue posible evaluar todos los enfoques, dado que no existen versiones públicas y/o disponibles de dichas herramientas (o existen errores en su instalación), como es el caso de los enfoques basados en bases de conocimiento o las redes neuronales basadas en LSTM.

Durante su comparación determinaron que tres de las herramientas basadas en CRF tienen el mejor rendimiento con un set de datos generado para resolver un caso con requerimientos definidos por ellos. Posteriormente, reentrenaron los modelos de estas tres herramientas, logrando mejorar su precisión y concluyendo que el CRF es el mejor de los enfoques que pudieron evaluar, principalmente por su capacidad de adaptarlo a requerimientos de extracción y estilos de citación diferentes.

Corpus

Los Corpus suelen ser recopilados por cada grupo de investigadores (en la mayoría de los artículos revisados), sin dar acceso a los mismos; sin embargo, existen algunas excepciones que utilizaron corpus de acceso libre y/o desarrollaron los propios y los hicieron públicos, la Tabla 2.1 recopila todos los mencionados en la literatura y sus características.

Tabla 2.1: Corpus empleados en la literatura relacionada con RM.

Conjunto de datos	N° de observaciones	Dominio
GROBID (constante crecimiento)	6835	Multidominio
Cora	1295	Ciencias de la computación
Citeseer	1563	Inteligencia Artificial
FLUX-CiM CS	300	Ciencias de la computación
FLUX-CiM HS	2000	Ciencias de la Salud
GROTOAP2	6858	Biomedicina, Ciencias de la computación
Venice	40000	Humanidades
GIANT	Un billón	Multidominio y multi estilos de citas

Grennan y cols. (2019) señalan que un sesgo significativo en la literatura es el uso de corpus con una cantidad insuficiente de registros, recomendando un mínimo de 8,000. Para abordar este problema, elaboraron un corpus de gran envergadura denominado Giant, que incluye cadenas de referencias etiquetadas en formato XML. El corpus abarca múltiples dominios y estilos de citación, y está disponible de manera gratuita para la comunidad investigadora.

Relación de fuentes citadas

En la Tabla 2.2 se puede apreciar una relación de todas las publicaciones utilizadas para construir el estado de arte, donde se enlistan notas, categorías, subtareas, y la métrica con la que se evaluó cada propuesta experimental.

Cabe aclarar que específicamente hablando de la columna **Evaluación,** el valor corresponde al promedio de los resultados ofrecidos por los autores en su propio conjunto de datos de entrenamiento.

Tabla 2.2: Tabla comparativa que resume las principales características de los artículos que componen el estado del arte.

Autor/es	Notas	Categoría	Subtareas	Evaluación
Choi y cols. (2023)	Bert Transfer Learning	Aprendizaje profundo	Segmentación	F1 0.9983
Uddin (2022)	Transformadores 220k referencias	Aprendizaje profundo	Segmentación	F1 0.942
Grennan y Beel (2020)	Referencias sintéticas	Conjuntos de datos	N/A	N/A
Yang y cols. (2020)	PBA - Generación de Plantillas - Meta-Learning	Otros	Segmentación	Exactitud 0.9833
Rizvi y cols. (2019)	DeepBird - Visión computacional	Aprendizaje profundo	Identificación y Extracción	AP50 98.56 %
Boukhers y cols. (2019)	Random Forest - CRF - Enfoque holístico	Aprendizaje máquina	Todas	Extracción F1-Score 0.78 Segmentación de F1-Score 0.92
Grennan y cols. (2019)	GIANT - Set de datos masivos	Conjuntos de datos		N/A
Prasad y cols. (2018)	Neural Parscit - BiLTSM - CRF	Aprendizaje profundo	Segmentación	F1-Score 0.9137
Rodrigues Alves y cols. (2018)	Bi-LTSM-CRF Embedings para preprocesamiento	Aprendizaje profundo	Todas	Extracción F1-Score 0.9509 Segmentación F1-Score 0.8966
Tkaczyk, Gupta, y cols. (2018)	ParsRec - Recomendación de analizadores Meta-Learning	Otros	Todas	F1-Score 0.891
Tkaczyk, Collins, y cols. (2018)	Comparativa	Comparativa	Segmentación	N/A
Körner (2017)	CRF - a líneas en vez de palabras	Aprendizaje máquina	Identificación y Extracción	N/A
Körner y cols. (2017)	CRF - a líneas en vez de palabras	Aprendizaje máquina	Identificación y Extracción	F1-Score 0.885
Bhardwaj y cols. (2017)	DeepBibx - Visión computacional	Aprendizaje profundo	Identificación y Extracción	Precisión 0.839 Recuerdo 0.846
Tkaczyk y cols. (2015)	CERMINE - SVM CRF K-Means - Heurística	Aprendizaje máquina	Todas	Extracción F1-Score 0.39 Segmentación F1- Score 0.89
Hsieh y cols. (2014)	FBA - enfoque basado en marcos - bases de conocimiento y FBA	Reglas y KB	Segmentación	Exactitud 0.9795
Ojokoh y cols. (2011)	HMM trigram	Aprendizaje máquina	Segmentación	F1-Score 0.8966
Zhang y cols. (2011)	SVM estructural	Aprendizaje máquina	Segmentación	Exactitud 0.9899
Zou y cols. (2010)	SVM y CRF comparación en HTML	Aprendizaje máquina	Segmentación	Exactitud 0.974
Romanello y cols. (2009)	CRF N-grams	Aprendizaje máquina	Identificación y Extracción	F1-Score 0.8707
Lopez (2009)	GROBID - CRF	Aprendizaje máquina	Segmentación	F1-Score 0.89
Hetzner (2008)	HMM de primer orden - referencias homogéneas	Aprendizaje máquina	Segmentación	F1-Score 0.847
Councill y cols. (2008)	ParsCit - CRF	Aprendizaje máquina	Segmentación	F1-Score 0.95
Day y cols. (2007)	Infomap - plantillas jerárquicas, estructura de árbol jerárquico, diversos estilos de citas	Plantillas	Segmentación	Exactitud 0.9239
Chen y cols. (2008)	BibPro – introducción de símbolos de puntuación para identificar el estilo de referencia – plantillas	Plantillas	Segmentación	F1-Score 0.9043
Cortez y cols. (2007)	Flux-CiM – KB, agnóstico del dominio	Reglas y KB	Segmentación	F1-Score 0.9639
Peng y McCallum (2006a)	CRF - variaciones gausianas	Aprendizaje máquina	Segmentación	F1-Score 0.915
Yin y cols. (2004)	HMM bigram	Aprendizaje máquina	Segmentación	Precisión 0.9015 Recuerdo 0.915
Ding y cols. (1999)	Múltiples plantillas para superar patrones irregulares.	Plantillas	Segmentación	gráficas imprecisas
Chowdhury (1999)	Heurística - reduce el error al proponer formato de llenado para escribir referencias.	Plantillas	Segmentación	N/A

2.3. Análisis del estado del arte

Para el sustento teórico de este proyecto doctoral se consultaron los buscadores de Google Scholar y Microsoft Academic, con la finalidad de identificar literatura relevante utilizando los términos "extracción de referencias, extracción de citas y análisis de referencias".

Se encontraron 114 artículos que hablan sobre el tema, de los cuales solamente veintinueve cumplieron con los criterios de inclusión, es decir, el artículo debe abordar una o más de las subtareas de Minería de referencias, el resto se descartó.

De las treinta publicaciones que cumplieron los criterios de inclusión, se pueden abstraer las siguientes categorías:

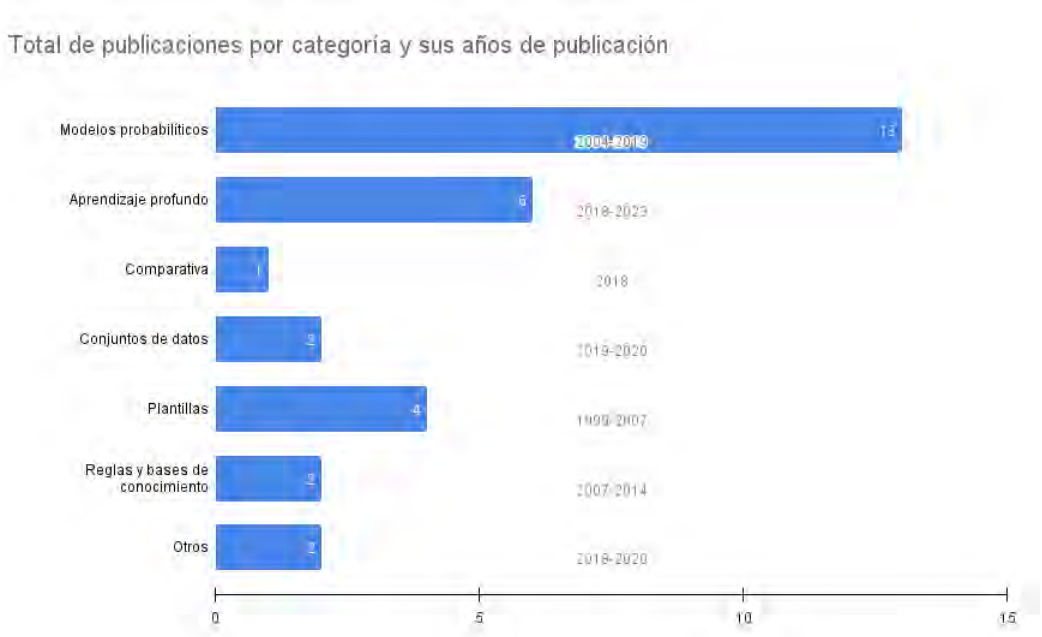


Figura 2.11: Categorías detectadas en el estado del arte.[2]

Observando la Figura 2.11: se pueden abstraer las siguientes conclusiones:

- Resalta el hecho de que el enfoque más popular es el de Aprendizaje máquina (13 publicaciones). Además, se distingue también por el hecho de ser el enfoque que más tiempo (quince años) se ha mantenido en el tema de RM.
- El enfoque basado en aprendizaje profundo tiene poco tiempo (empezó en 2018) y pocas publicaciones (5).
- El enfoque basado en plantillas fue de los primeros (nació en 1999) pero empezó a dejar de utilizarse en 2007.

[2]KB corresponde a bases de conocimiento.

- Se detectaron dos enfoques alternativos. El primero es de Yang y cols. (2020) quienes lo llaman "Meta-aprendizaje", que consiste en el uso de aprendizaje automático para encontrar/diseñar plantillas (reviviendo dicho enfoque) que puedan ajustarse a cada referencia que se quiera identificar, extraer y segmentar, en tanto que la otra propuesta (Tkaczyk, Gupta, y cols., 2018) consiste en un algoritmo que recomienda un segmentador de referencias idóneo para cada caso.
- Comparativas entre enfoques, solamente ha surgido una, y no tuvo los medios para evaluar aprendizaje profundo.
- Solamente se encontraron dos publicaciones que emplean el enfoque de reglas y bases de conocimiento, pero dejaron de utilizarse en 2014.
- Cabe mencionar que el tema de RM ha suscitado la aparición una publicación que trata específicamente tratan sobre el tema de los Corpus, donde en la primera se propone un set con más de un billón de observaciones (Grennan y cols., 2019). Y en la segunda se explora la posibilidad de crear "citas sintéticas", concluyendo que son efectivas para entrenar modelos basados en DL (Grennan y Beel, 2020).

Tabla 2.3: Relación de trabajos revisados y Factores de Contexto, Subtareas, Idioma, Métricas y Dominio.

Factor	Valor	Cantidad	Autores
Idioma	Alemán	1	Körner y cols. (2017)
	Inglés	28	Restantes
	Inglés/Coreano	1	Choi y cols. (2023)
Dominio	Humanidades	1	Rodrigues Alves y cols. (2018)
	Ciencias de la Computación/ Ciencias de la Salud	27	Restantes
	Multidominio	2	Uddin (2022) Choi y cols. (2023)
Contexto	Imagen	2	Rizvi y cols. (2019) Bhardwaj y cols. (2017)
	Texto	28	Restantes
Subtareas	Identificación, Extracción y Segmentación	4	Tkacyzk et al. 2015 Boukhers y cols. (2019) Yang y cols. (2020) Choi y cols. (2023)
	Identificación y Extracción	4	Rizvi y cols. (2019) Bhardwaj y cols. (2017) Körner (2017) Körner y cols. (2017)
	Segmentación	22	Restantes

En la Tabla 2.3, se pueden observar aspectos distintivos de las publicaciones revisadas dentro del estado del arte. Llegando a las siguientes afirmaciones:

- Prácticamente, todas las publicaciones trabajan con referencias escritas en idioma inglés (93.33 %).
- Prácticamente, todas las publicaciones trabajan con referencias que pertenecen al dominio de Ciencias computacionales y/o Ciencias de la salud, los cuales tienen un estilo de citación muy uniforme (90 %).
- Segmentación es la subtarea más estudiada (73.33 %).
- La información de contexto más empleada es el texto (93.33 %).
- La información de contexto basada en imagen (6.9 %) solamente se emplea para las tareas de Identificación y Extracción. Son escasas las publicaciones que abordan todo el proceso de RM (13.33 %).

A manera de conclusión, se pueden apreciar que existen una serie de omisiones en la literatura con respecto a dos aspectos:

- El idioma (predominio del inglés)
- El dominio (predominio de los dominios de ciencias de la salud y ciencias computacionales)

Si realmente se espera que la Minería de referencias tenga un verdadero impacto en el quehacer científico, es imperioso comenzar a expandir las fronteras en la aplicación de las técnicas ya desarrolladas, evaluarlas en nuevos escenarios (como nuevos dominios y/o lenguajes) y en caso de detectar problemas desarrollar estrategias que permitan superarlos.

2.4. Análisis de la segmentación de referencias

La segmentación de referencias bibliográficas es una tarea especializada dentro del campo del procesamiento del lenguaje natural. Consiste, por lo general, en tres fases interconectadas que facilitan la identificación y clasificación de los componentes de una referencia bibliográfica, tales como autores, títulos y datos de publicación.

La primera fase implica la selección de características relevantes del texto. El objetivo es identificar y extraer características que son relevantes de cada token de una referencia, obteniendo información relevante para las siguientes fases. La selección es fundamental porque las características adecuadas pueden determinar significativamente la eficacia de las fases subsiguientes.

Una vez que se han seleccionado las características, la segunda fase del proceso consiste en capturar el contexto en el que los tokens aparecen. Este paso es esencial para entender cómo se relacionan los diferentes componentes de una referencia entre sí. La capacidad de analizar el contexto permite al sistema hacer inferencias más precisas sobre la estructura y la clase a la que pertenece cada token de una referencia bibliográfica.

Finalmente, la tercera fase es la de predicción, donde se utilizan los datos preparados y analizados en las fases anteriores para clasificar cada elemento identificado en las categorías correspondientes. Es decir, es el punto donde el sistema decide, basándose en la información y el contexto capturados, cómo etiquetar cada parte de la referencia bibliográfica.

El resultado es una segmentación precisa de las referencias, lo que permite su posterior uso en sistemas de gestión de información bibliográfica y análisis académico.

En conjunto, las tres fases descritas constituyen un enfoque integral para la segmentación de referencias bibliográficas, proporcionando una metodología robusta.

A continuación se describe, con mayor detalle, cada fase y se hará referencia a los artículos más representativos dentro de la literatura.

2.4.1. Fase 1: Selección de Características

La selección de características es un paso determinante en las arquitecturas para la segmentación de referencias bibliográficas. La elección de las características adecuadas impacta directamente en la eficacia y eficiencia del modelo final. Dentro del estado del arte, se han encontrado diversas formas de realizar la selección de características, mismas que pueden ser agrupadas en dos categorías que se describen más adelante.

Selección manual de características

En modelos basados en ML clásico, la selección de características se realiza manualmente. Lo que implica un análisis detallado y específico del dominio para identificar patrones gramaticales, lexicográficos y semánticos útiles.

Entre las características relevantes se incluyen la capitalización de las palabras, uso de comas, entre otros, que son indicativos de cada token de componente de una referencia bibliográfica, los ejemplos más representativos (y con mejores resultados) de la literatura son ParsCit (Councill y cols., 2008) y GROBID (Lopez, 2009) en su modalidad CRF[3].

Cabe aclarar que las implementaciones que usan ML clásico solamente contemplan dos fases (selección de características y predicción).

Selección automática de características

En modelos basados en DL, la selección de características se realiza de manera automática. Es decir, algoritmos con diferentes características extraen información de cada token y generan una representación vectorial del mismo, cuantificando relaciones sintácticas y semánticas.

Los algoritmos utilizados en la literatura son los siguientes:

- Incrustaciones de Palabras: Modelos como Word2Vec (Lopez, 2009; Prasad y cols., 2018; Rodrigues Alves y cols., 2018; Uddin, 2022), y ELMo (Lopez, 2009) permiten capturar el contexto y la semántica de las palabras dentro de las referencias bibliográficas. Los *embeddings* generan representaciones vectoriales de palabras basadas en grandes corpus de texto, lo que ayuda a capturar relaciones semánticas complejas.

[3]Cabe aclarar que GROBID es un sistema que se ha mantenido evolucionando e incluye multitud de herramientas para la extracción de información de documentos académicos, en el caso de la segmentación de referencias, implementa CRF Clásico, BiLSTM con *embeddings* ELMo (Peters y cols., 2018) y Word2Vec

- Incrustaciones de caracteres: La representación a nivel de caracteres permite modelar subcomponentes de las palabras, útil especialmente para capturar prefijos, sufijos y la estructura interna de términos técnicos o nombres propios que son frecuentes en referencias bibliográficas.

Una particularidad de los enfoques basados en DL, encontrados en la literatura, es que utilizan ambos algoritmos de incrustaciones, es decir concatenan incrustaciones de palabras y de caracteres en un solamente vector.

2.4.2. Fase 2: Captura del Contexto

Una vez que las representaciones vectoriales de los tokens están definidas, la siguiente fase (exclusiva de las arquitecturas basadas en DL) para la segmentación de referencias bibliográficas es la captura del contexto.

La fase de captura de contexto es fundamental para entender las dependencias y relaciones entre los tokens dentro de una referencia bibliográfica, lo que permite identificar y clasificar correctamente las diferentes partes de una referencia (como autores, título, año de publicación, etc.).

Las técnicas empleadas en el estado del arte son las siguientes:

LSTM (redes neuronales recurrentes de Memoria a Largo y Corto Plazo)

Las redes LSTM (Long Short-Term Memory) son una clase de redes neuronales recurrentes diseñadas específicamente para evitar el problema del desvanecimiento del gradiente que afecta a las redes recurrentes estándar (Hochreiter y Schmidhuber, 1997).

Las LSTM son particularmente efectivas en el contexto de la segmentación de referencias bibliográficas, las LSTM pueden capturar información relevante de los tokens previos para ayudar a determinar la naturaleza de los tokens subsiguientes, facilitando la identificación precisa de las partes constituyentes de una referencia, como nombres de autores o títulos de trabajos(Prasad y cols., 2018).

Por otra parte, las redes BiLSTMs (Bidirectional Long Short-Term Memory) extienden la funcionalidad de las LSTM al procesar los datos en ambas direcciones, tanto del pasado hacia el futuro como del futuro hacia el pasado, en paralelo (Graves y Schmidhuber, 2005).

El procesamiento bidireccional permite capturar el contexto de cada token tanto desde lo que precede como lo que sigue en el texto. Al incorporar información de ambos lados de un token, las BiLSTMs ofrecen un contexto aún más amplio de cada token al momento de realizar la segmentación de referencias, siendo Prasad y cols. (2018) y Rodrigues Alves y cols. (2018) quienes la utilizan.

Codificador de tipo Transformador

Los codificadores de tipo Transformador utilizan mecanismos de atención para ponderar la importancia relativa de cada token en relación con los demás dentro de la secuencia.

El codificador de un Transformador no procesa secuencialmente el texto, sino en paralelo, mediante un codificador de posiciones, que genera una matriz que pasa a través de mecanismos de atención multicabeza, capturando relaciones entre tokens distantes de manera eficaz (Vaswani y cols., 2017).

La capacidad para gestionar dependencias a largo plazo es significativamente útil en la segmentación de referencias bibliográficas, donde elementos como el nombre del autor, el título del trabajo y otros detalles pueden estar separados por múltiples tokens (dependiendo del estilo).

Dentro de la literatura, Uddin (2022) se distingue por implementar un codificador de un Transformador para capturar el contexto de las referencias bibliográficas.

2.4.3. Fase 3: Predicción

La predicción es la etapa final en las arquitecturas de ML aplicadas a la segmentación de referencias bibliográficas. Aprovecha toda la información procesada y contextualizada en las fases anteriores para clasificar cada token, asignándolo a una categoría específica (como nombre de autor, título del artículo, fecha de publicación, etc.).

La fase de predicción es fundamental para convertir las representaciones y contextos capturados en etiquetas concretas que definen la estructura de una referencia.

Los CRF (Lafferty y cols., 2001) son especialmente populares en tareas de NER y dominan en el campo de la segmentación de referencias bibliográficas[4]. Su eficacia radica en la modelación de las dependencias secuenciales en los datos etiquetados, dado que la categoría de un token no solamente depende de las características del token en sí, sino también de las etiquetas asignadas a los tokens adyacentes.

Los CRF aportan coherencia y precisión de las etiquetas en secuencias donde las relaciones entre elementos son complejas y dependientes del contexto.

2.4.4. Discusión

En el análisis de la segmentación de referencias bibliográficas, las fases descritas constituyen un marco robusto para la estructuración y el entendimiento de textos académicos y científicos. Sin embargo, el proceso no está exento de desafíos, especialmente cuando se trata de aplicarse en idiomas con recursos limitados o en formatos de referencia altamente variables (Jain y cols., 2023).

Es importante destacar que mientras las implementaciones basadas en aprendizaje profundo ofrecen ventajas significativas en términos de automatización y precisión, también enfrentan desafíos, especialmente relacionados con la dependencia de recursos lingüísticos, los cuales están desarrollados principalmente para el inglés (Boukhers y cols., 2019).

Muchas de las mencionadas implementaciones requieren corpus específicos que no siempre están disponibles para pruebas externas, limitando su generalización en diferentes contextos lingüísticos y académicos.

Las técnicas de aprendizaje profundo, tales como LSTM, BiLSTM, y Transformers, han mostrado ser particularmente efectivas para capturar contextos complejos y relaciones de largo alcance entre los componentes de las referencias bibliográficas. Además, automatizan la selección de características y mejoran la capacidad del modelo para adaptarse a nuevos patrones sin intervención manual extensiva.

Sin embargo, la efectividad de los enfoques basados en DL está intrínsecamente ligada a la disponibilidad de grandes corpus anotados y la capacidad de generalizar y ser eficiente en otros conjuntos de

[4](por sus resultados por encima de algoritmos como SVM, Softmax y HMM

datos (Grennan y Beel, 2020). Convirtiéndose en un punto de fricción significativo cuando se intenta aplicarlos a idiomas o disciplinas para los cuales no existen suficientes datos de entrenamiento.

El dominio del inglés en los recursos y herramientas de procesamiento del lenguaje natural también plantea limitaciones significativas. Muchas de las implementaciones más avanzadas están diseñadas y optimizadas para inglés, lo que limita su generalización directa en otros idiomas. No solamente afectando la accesibilidad de las tecnologías de segmentación de referencias para investigadores en otros idiomas, sino que también puede sesgar los avances tecnológicos al centrarlos en el inglés, dejando de lado el universo de publicaciones académicas escritas en otros idiomas.

Para abordar dichos desafíos y mejorar la segmentación de referencias bibliográficas en múltiples idiomas, es esencial considerar dos áreas clave de desarrollo. En primer lugar, la selección de características debe ser adaptada y enriquecida para abarcar las variaciones lingüísticas específicas de diferentes idiomas. Lo que implica la búsqueda o desarrollo de recursos lingüísticos adecuados para el propósito enunciado.

En segundo lugar, mejorar la captura de contexto en los modelos podría significativamente incrementar su tolerancia a errores y su capacidad para gestionar incoherencias en los datos de entrada. Dichas mejoras no solamente ayudarían a comprender el contexto en que se insertan las referencias bibliográficas, sino que también permitirían manejar de manera más efectiva las irregularidades y las variaciones estructurales que frecuentemente presentan los documentos académicos.

Implementando tales mejoras, se espera que los sistemas de segmentación de referencias bibliográficas sean más robustos, versátiles y accesibles para una comunidad global de investigadores, mejorando así la eficacia con la que se gestionan y utilizan las referencias bibliográficas en el ámbito académico y científico.

3 Propuesta para mejorar la generalización de la segmentación de referencias a más idiomas y estilos

A lo largo del presente capítulo se presenta una propuesta de mejora para la subtarea de segmentación de referencias bibliográficas. El objetivo es extender la generalización del proceso a múltiples idiomas y estilos, y minimizar el impacto de errores. La propuesta contempla las siguientes estrategias:

1. Construcción de un corpus de referencias que comprenda un balance y suficiencia representativa de la mayor cantidad de estilos de referencias posible.
2. Ajustes en la Selección de Características: Implementación de modificaciones en los recursos lingüísticos utilizados actualmente, con el objetivo de refinar las limitaciones encontradas en el estado del arte con respecto al idioma.
3. Selección de la arquitectura óptima para la fase de Captura de Contexto: Evaluación comparativa entre diversas arquitecturas existentes para identificar aquella que demuestre una mayor tolerancia a errores e irregularidades.

A continuación, se detallan las estrategias propuestas:

3.1. Construcción del corpus de Referencias

Para solventar el requerimiento de construir un modelo capaz de segmentar referencias en diferentes estilos, se utilizó el corpus GIANT (Grennan y cols., 2019), reconocido por su amplia variedad de estilos de referencias aplicados en distintos tipos de documentos académicos.

A partir de GIANT se extrajo un conjunto de datos de referencias diversificado y representativo. El conjunto se distribuye en subconjuntos para las fases de entrenamiento, ajuste, evaluación y validación. Cada subconjunto está concebido para ser una muestra equilibrada y representativa, garantizando una cobertura exhaustiva de los variados estilos y tipos de documentos académicos.

Con el fin de evaluar la capacidad del modelo para segmentar referencias en múltiples idiomas, se construyó un corpus llamado Redalyc, que incluye referencias en inglés, español y francés, facilitando así la evaluación del modelo en un contexto multilingüe.

Para probar la tolerancia del sistema a las inconsistencias en las referencias, se utilizó el corpus CORA (Anzaroot y McCallum, 2013). El conjunto de datos es relevante para evaluar cómo el sistema

maneja las irregularidades y errores que frecuentemente se presentan en las referencias bibliográficas reales. La inclusión del corpus CORA es fundamental para asegurar que el sistema pueda funcionar de manera robusta bajo condiciones no adecuadas (para información más detallada ver sección 3.4.1).

La estrategia de construcción, descrita anteriormente, del corpus tiene como objetivo que el modelo tenga un adecuado desempeño bajo un espectro amplio de escenarios y aumentar su robustez y precisión en la segmentación de referencias bibliográficas.

3.2. Mejoras en la selección de características

Se proponen ajustes específicos destinados a ampliar la capacidad del sistema para procesar y entender referencias bibliográficas en un rango más amplio de idiomas.

Tal como se menciona en la sección 2.4.1, la selección automática de características se realiza mediante la concatenación de incrustaciones de palabras e incrustaciones de caracteres.

Las incrustaciones de palabras, son un enfoque fundamental para capturar relaciones semánticas y sintácticas a nivel de palabras, mediante el uso de modelos como Word2Vec (Mikolov y cols., 2013) o ELMo (Peters y cols., 2018).

Dichas incrustaciones transforman palabras en vectores densos en un espacio continuo. Una característica notable de las incrustaciones de palabras es su capacidad para captar relaciones y analogías entre palabras, basándose en la co-ocurrencia y la proximidad contextual dentro de grandes corpus de texto.

Aunque tienen limitaciones para manejar palabras nuevas o desconocidas fuera del vocabulario inicial, su fortaleza radica en consolidar un conocimiento semántico profundo que beneficia múltiples aplicaciones en el procesamiento del lenguaje natural, desde la búsqueda semántica hasta la clasificación de texto, entre otras tareas. Los vectores resultantes ofrecen una representación que refleja las similitudes y diferencias en el uso lingüístico a lo largo de amplios contextos (Almeida y Xexéo, 2019).

Por otro lado, las incrustaciones de caracteres, son un enfoque particularmente útil para capturar información morfológica y sintáctica a nivel más granular, el cual emplea una red BiLSTM para convertir caracteres en vectores. Una particularidad de las incrustaciones de caracteres es que son capaces de generar representaciones para palabras nuevas o desconocidas (Ling y cols., 2015).

En el ámbito de la segmentación de referencias, el uso de modelos de incrustaciones de palabras preentrenados se presenta como una estrategia efectiva para aprovechar el conocimiento acumulado. Los modelos de incrustaciones de palabras, como Word2Vec o ELMo, a menudo se basan en un vocabulario predefinido, limitando su capacidad para gestionar términos nuevos o desconocidos.

No obstante, la combinación de incrustaciones de palabras con incrustaciones de caracteres suelen utilizarse en la literatura como una solución viable. Mientras las incrustaciones de palabras capturan contextos semánticos amplios, las incrustaciones de caracteres ofrecen flexibilidad al representar palabras a partir de sus unidades más básicas, lo que permite generar representaciones para términos no incluidos en el vocabulario inicial.

En contextos multilingües, el desafío se intensifica debido a la diversidad lingüística y la variabilidad morfológica entre los idiomas. Los modelos que dependen de un vocabulario fijo suelen encontrarse limitados cuando se enfrentan a términos nuevos o especializados, especialmente en campos como el científico y técnico, donde la terminología evoluciona y cambia rápidamente.

En la segmentación de referencias dentro de textos académicos, el problema antes mencionado es aún más evidente. Las referencias poseen una estructura sintáctica compleja y una carga semántica que varía significativamente, complicando la identificación y el procesamiento automático de los términos utilizados.

Para abordar retos, se propone reemplazar, en la fase de selección de características, a las incrutaciones de palabras por el modelo multilingüe BPEmb (Heinzerling y Strube, 2018), que comprende una colección de incrustaciones subléxicas preentrenadas en 275 idiomas.

BPEmb se basa en la técnica de Byte Pair Encoding (BPE), optimizando el manejo de palabras y términos que frecuentemente están fuera del vocabulario en modelos convencionales. Mediante la fusión repetida de los pares de caracteres más frecuentes en un corpus, BPE crea subunidades de texto que pueden representar eficientemente fragmentos de palabras o palabras completas, facilitando así el manejo de términos nuevos o especializados.

Al descomponer palabras en subunidades comunes, BPE ofrece un enfoque más uniforme para el procesamiento de múltiples idiomas, incluidos aquellos con sistemas de escritura no latinos.

La integración de BPEmb junto con incrustaciones de caracteres busca ofrecer una solución robusta y flexible para la segmentación de referencias bibliográficas multilingües. Además, BPEmb se destaca por su eficiencia en la utilización de recursos computacionales, ofreciendo una solución viable para la implementación en sistemas con restricciones de recursos, como dispositivos móviles.

3.3. Selección de la arquitectura óptima para la fase de captura de contexto

Se realizaron evaluaciones de diferentes arquitecturas para la fase de Captura de contexto, con el objetivo de mejorar la generalización de la segmentación de referencias, y aumentar la precisión y robustez del modelo, y a su vez, asegurar la eficacia en un entorno académico global y diverso.

El principal desafío radica en la complejidad sintáctica predominante en las referencias bibliográficas. Estructuras, que incluyen elementos como nombres de autores, títulos de trabajos y detalles de publicación, presentan retos significativos en la segmentación y reconocimiento de términos.

Para enfrentar dichos obstáculos, se plantea una evaluación comparativa de arquitecturas avanzadas en el procesamiento del lenguaje natural, destacando el uso de BiLSTM y codificadores de Transformadores, referenciados en estudios como Prasad y cols. (2018); Rodrigues Alves y cols. (2018); Uddin (2022).

Se compararon mencionadas arquitecturas no solamente para contrastar su eficacia en la gestión de referencias con inconsistencias, sino también para determinar combinaciones que mejoren su rendimiento. El fin es adaptar tales arquitecturas para que gestionen con eficiencia la variedad de estructuras sintácticas que se utilizan para estructurar las referencias bibliográficas.

Se debe aclarar que no se contempló el uso de BERT, a pesar de su eficacia mostrada en múltiples tareas de NLP, dado que no se considera viable debido a su elevado costo computacional, especialmente para tareas de segmentación como se discute en Choi y cols. (2023). Se prioriza mantener una eficiencia computacional que no comprometa la capacidad de procesamiento requerida.

Para validar el éxito de las arquitecturas propuestas se construyeron los siguientes modelos:

1. Sin captura de contexto (en adelante, modelo CRF).
2. BiLSTM (modelo BiLSTM).
3. Word2Vec + BiLSTM (modelo Word2Vec + BiLSTM).
4. Codificador de Transformador (modelo Transformer).
5. Codificador de Transformador con Atención personalizada (modelo Sliding Window).
6. Combinación de Codificador de Transformador y BiLSTM (modelo Transformer + BiLSTM).
7. Integración de Codificador de Transformador con Atención personalizada y BiLSTM(modelo Sliding Windows + BiLSTM).

La presente propuesta busca establecer un enfoque más preciso y adaptable para la segmentación de referencias, capaz de acomodar la diversidad terminológica y estructural propia de múltiples idiomas y campos del saber.

A continuación, se muestra un esquema general (ver Figura 3.1) que refleja el flujo de trabajo implementado. El cual sigue el proceso típico del estado del arte y destaca las combinaciones propuestas. Cabe señalar que todos los modelos fueron construidos desde cero como parte de este proyecto doctoral para abordar la tarea específica de la segmentación de referencias.

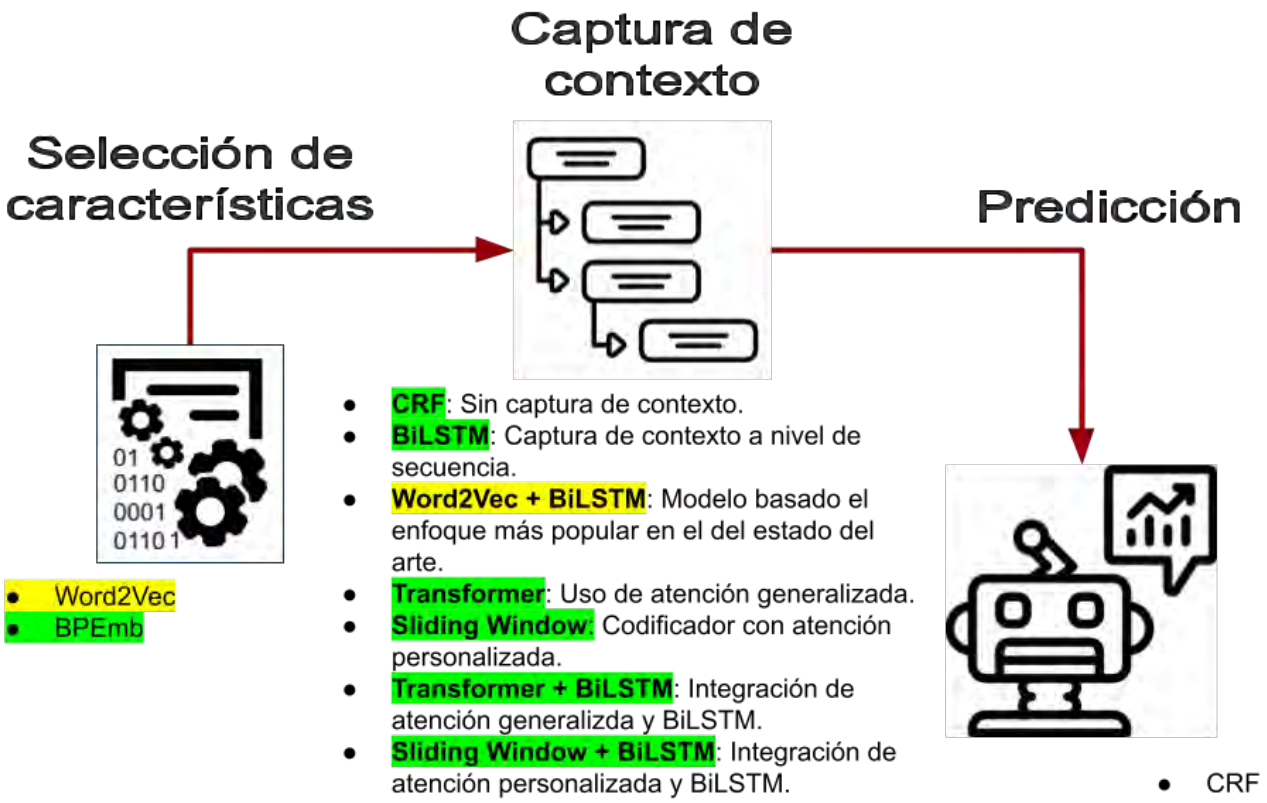


Figura 3.1: Esquema general de la solución, se muestran las fases y las variaciones que se implementaron (iconos obtenidos en: https://www.pngegg.com/es).

El flujo de trabajo consiste en combinar una técnica de selección de características con una arquitectura de captura de contexto, generando así un total de ocho modelos distintos. Las combinaciones empleadas se distinguen en el esquema mediante los colores amarillo y verde. El color amarillo representa la combinación que emula las estrategias más populares del estado del arte en DL (ver sección 2.4), mientras que el color verde corresponde a otras combinaciones propuestas.

Entre las arquitecturas evaluadas se incluyen modelos como CRF, BiLSTM, Transformer, y variantes más avanzadas que integran Sliding Window con BiLSTM. Las arquitecturas basadas en Sliding Window fueron diseñadas como propuestas originales en este proyecto (para información más detallada ver las secciones 3.6.5 y 3.6.7), con el objetivo de adaptar el procesamiento de contexto de manera más acorde con la complejidad y naturaleza secuencial de la tarea de segmentación de referencias.

Este enfoque permite capturar información relevante tanto a nivel morfológico como semántico, asegurando que el modelo sea capaz de manejar múltiples idiomas y estilos, así como adaptarse a las variaciones estructurales que presentan las referencias bibliográficas en diferentes contextos.

3.4. Método de solución

A lo largo de la presente sección se detallan las estrategias descritas en apartados anteriores y los experimentos realizados para validar su funcionamiento.

3.4.1. Adquisición de datos (construcción del corpus de referencias)

En esta sección se describen los métodos que se utilizaron para generar/obtener los tres corpus utilizados para el entrenamiento y experimentación del modelo, así como los procesos realizados para obtener muestras representativas y el proceso de limpieza utilizado.

Para los fines del presente trabajo se utilizaron dos corpus obtenidos del estado del arte y se construyó uno adicional, sus particularidades se describen a continuación.

Corpus GIANT

El corpus Giant[1], construido por Grennan y cols. (2019), cuenta con 991.411.110 referencias, las cuales están divididas en 1568 estilos diferentes de citación y contemplan 24 tipos de documentos, cada registro cuenta con la siguiente estructura:


```
{
 "doi": "10.2307/2177340",
 "articleType": 3,
 "citationStyle": 0,
 "citationStringAnnotated": "<author><family>Ritchie</family>, <
    given>E.</given> and <family>Powell</family>, <given>Elmer
    Ellsworth</given></author> (<issued><year>1907</year></issued>)
     <title>Spinoza and Religion.</title> <container-title>The
```


[1] https://doi.org/10.7910/DVN/LXQXAO

```
    Philosophical Review</container-title>, <volume>16</volume>(<
    issue>3</issue>), p. <page>339</page>. [online] Available from:
     <URL>http://dx.doi.org/10.2307/2177340</URL>"
}
```

Cada campo contiene la siguiente información:

- **doi**: El id único del documento al que representa [2].
- **articleType**: El id que representa el tipo de documento (tesis, artículo, libro, etc).
- **citationStyle**: El id que representa el estilo de citación.
- **citationStringAnnotated**: La cadena de referencia etiquetada.

La cadena de referencia viene etiquetada con la siguiente estructura en XML:

```
<author>
<family>Ritchie</family>,
<given>E.</given> and
<family>Powell</family>,
<given>Elmer Ellsworth</given>
</author> (
<issued>
<year>1907</year>
</issued>)
<title>Spinoza and Religion.</title>
<container-title>The Philosophical Review</container-title>,
<volume>16</volume>(
<issue>3</issue>), p.
<page>339</page>. [online] Available from:
<URL>http://dx.doi.org/10.2307/2177340</URL>
```

Debido a que el corpus presenta muy pocas anotaciones sobre las características de cada registro (solamente se incluye el DOI, estilo de citación y tipo de documento), se tomó la decisión de identificar el idioma de cada registro, con la finalidad de construir el corpus en español, para lo que se llevó a cabo el siguiente procedimiento:

1. Se decidió tomar el título como único elemento que aporta certeza sobre el idioma de la publicación.
2. Se extrae el título de la estructura XML almacenada en el campo citationStringAnnotated de cada registro.
3. Se somete a identificación mediante cinco métodos de detección del idioma.
4. Se toma como correcto el título que presente mínimo tres coincidencias del idioma detectado.

[2] https://ask.library.uic.edu/faq/345899

Los cinco métodos utilizados para la detección de idioma son:

- Spacy LangDetector[3]
- gcld3[4]
- google_trans_new[5]
- El sitio `https://detectlanguage.com/`, se construyó un script que automatiza el proceso de consultar el formulario del demo dado que la API es de pago.
- El sitio `https://www.cortical.io/freetools/detect-language/`, se construyó un script que automatiza el proceso de consultar el formulario del demo dado que la API es de pago.

Cabe señalar que únicamente se identificaron los registros de un solamente estilo de referencia (APA en este caso), dado que determinar el idioma de todos los registros de un estilo es fácilmente transferible a los otros mediante el campo DOI.

Finalmente, cabe destacar que, gracias al proceso de identificación de idiomas previamente mencionado, se estima que las referencias en francés en GIANT representan el 3.5 %, mientras que las referencias en español constituyen el 0.77 %

Corpus CORA

Se trata de un corpus anotado por humanos de 1877 cadenas de referencias bibliográficas con una variedad de formatos y estilos, incluyendo preimpresiones de revistas, ponencias de conferencias e informes técnicos[6] construido y actualizado (Peng y McCallum, 2006b; Anzaroot y McCallum, 2013), el cual es muy utilizado con propósitos de evaluación de la subtarea de segmentación de referencias (Councill y cols., 2008; Prasad y cols., 2018; Grennan y Beel, 2020).

La cadena de referencia viene etiquetada con la siguiente estructura:

```
<NEWREFERENCE>
ahlskog1994a <author> M. Ahlskog, J. Paloheimo, H. Stubb, P.
   Dyreklev, M. Fahlman, O. </author> <title> Inganas and M.R. </
   title> <journal> Andersson, J Appl. Phys., </journal> <volume>
   76, </volume><pages>893,</pages> <date> (1994). </date>
```

Corpus Redalyc

Dado que el idioma de las referencias es uno de los grandes faltantes en la literatura (por ejemplo GIANT cuenta con muy pocos ejemplos diferentes al inglés), se tomó la decisión de crear un corpus con el idioma etiquetado, con la finalidad de evaluar las capacidades multilingües de los modelos construidos.

[3]https://github.com/Abhijit-2592/spacy-langdetect
[4]https://pypi.org/project/gcld3/
[5]https://pypi.org/project/google-trans-new/
[6]https://people.cs.umass.edu/~mccallum/data/cora-refs.tar.gz

Como primera acción, se realizó la consulta de diversas bases de datos (MS académico, Google Scholar, CiteSeerX, sCielo, etc.) buscando que cuenten con la opción de filtrar por idioma, resultando que la única con tal característica fue Redalyc[7].

Redalyc.org permite realizar consultas de artículos en diversos idiomas, además de que provee la cadena Bibtex[8] de cada uno, también provee la forma de citar (una cadena de texto) cada artículo en siete estilos diferentes (APA6, MLA, Chicago, Vancouver, Harvard, ISO NLM).

Dado que el sitio no ofrece una API de consulta, se construyó un extractor (webscrapper en inglés) empleando el lenguaje Python. Tal extractor realiza las búsquedas y aplica los filtros correspondientes (idioma), utilizando la librería Selenium. El proceso descrito como algoritmo es:

1. **Inicialización**:
   - Configurar el entorno de Selenium con Google Chrome.
   - Establecer los parámetros de búsqueda (filtros por idioma).
2. **Acceso y navegación en Redalyc**:
   - Abrir el sitio web de Redalyc.
   - Ingresar la consulta de artículos según los filtros especificados (por idioma).
3. **Manejo de paginación y carga dinámica**:
   - Realizar la primera búsqueda y esperar la carga completa de los resultados mediante Ajax.
   - Implementar pausas estratégicas en el código para garantizar la carga de los datos.
4. **Extracción de datos**:
   - Recorrer cada página de resultados.
   - Extraer la cadena `Bibtex` de cada artículo.
   - Extraer las cadenas de referencia en diferentes estilos (APA6, MLA, Chicago, etc.).
   - Navegar a la siguiente página de resultados y repetir el proceso hasta el final.
5. **Almacenamiento**:
   - Guardar los datos en archivos separados por idioma y estilo en formato CSV o JSON.
6. **Finalización**:
   - Cerrar el navegador y finalizar el proceso de scraping.

Selenium automatiza acciones dentro de un navegador (en este caso Google Chrome). Para sortear la primera dificultad (pausando la ejecución del código en diferentes puntos para poder garantizar la carga de los datos en el navegador), la cual consiste en que toda la búsqueda y paginación se lleva a cabo mediante llamadas Ajax.

De esta forma se logró extraer las cadenas Bibtex, las cadenas de referencias crudas en los estilos antes mencionados y construir tres conjuntos de datos en idiomas etiquetados:

[7] https://www.redalyc.org/
[8] https://www.bibtex.org/

1. Redalyc_spa (Español) con 346K registros.
2. Redalyc_eng (Inglés) con 98K registros.
3. Redalyc_fra (Francés) con 2K registros.

Es importante mencionar que los tres idiomas son los más comunes en redalyc.org, a pesar de que existen otros idiomas como italiano o alemán, la cantidad de referencias es significativamente baja.

En la figura 3.2 se puede apreciar la estructura de un registro del corpus Redalyc.


```
{
  "_id": {
    "$oid": "6186a0bb8bc7ea515a91a0ed"
  },
  "link": "https://www.redalyc.org/articulo.oa?id=366638654012",
  "apa6": "Núñez, A. (2005). Cartas a la Dirección. Pediatría Atención Primaria, VII(25),139-141.[fecha de Consulta 6 de Noviembre de 2021]. ISSN: 1139-7632. Disponible en: https://www.redalyc.org/articulo.oa?id=366638654012",
  "chicago": "Núñez, A., y \"Cartas a la Dirección.\" Pediatría Atención Primaria VII, no. 25 (2005):139-141. Redalyc, https://www.redalyc.org/articulo.oa?id=366638654012",
  "harvard": "Núñez, A. y (2005), \"Cartas a la Dirección.\" Pediatría Atención Primaria, Vol. VII, núm.25, pp.139-141 [Consultado: 6 de Noviembre de 2021]. ISSN: 1139-7632. Disponible en : https://www.redalyc.org/articulo.oa?id=366638654012",
  "iso": "Núñez, A. Cartas a la Dirección. Pediatría Atención Primaria [en línea]. 2005, VII(25), 139-141[fecha de Consulta 6 de Noviembre de 2021]. ISSN: 1139-7632. Disponible en: https://www.redalyc.org/articulo.oa?id=366638654012",
  "mla": "Núñez, A., y \"Cartas a la Dirección.\" Pediatría Atención Primaria, vol. VII, no. 25, 2005, pp.139-141. Redalyc, https://www.redalyc.org/articulo.oa?id=366638654012",
  "nlm": "Núñez, A. Cartas a la Dirección. Pediatría Atención Primaria. 2005;VII(25):139-141.[fecha de Consulta 6 de Noviembre de 2021]. ISSN: 1139-7632. Disponible en: https://www.redalyc.org/articulo.oa?id=366638654012",
  "vancouver": "Núñez, A., Cartas a la Dirección. Pediatría Atención Primaria [Internet]. 2005;VII(25):139-141. Recuperado de: https://www.redalyc.org/articulo.oa?id=366638654012",
  "bibtex": "@article{366638654012,\n abstract = {},\n author = {Núñez, A.},\n issn = {1139-7632},\n journal = {Pediatría Atención Primaria},\n keywords = {},\n language = {Español},\n number = {25},\n pages = {139-141},\n title = {Cartas a la Dirección},\n url = {https://www.redalyc.org/articulo.oa?id=366638654012},\n volume = {VII},\n year = {2005}\n}\n",
}
```


Figura 3.2: Registro de ejemplo corpus Redalyc.

3.4.2. Preprocesamiento

Preprocesamiento GIANT

Para fines del presente trabajo, se procedió a simplificar la referencia etiquetada para que solamente se mantengan las siguientes etiquetas, las cuales son consideradas el mínimo necesario para identificar una obra.

Quedando de la siguiente manera:

-Author	- Year	-Title	-Container-Title
-Volume	-Issue	-Page	-ISBN
-ISSN	-Publisher	-DOI	-URL

Ejemplo de registro en Giant:

```
<author>
Ritchie E. and Powell, Elmer Ellsworth
</author>
(
```

```
<year>
  1907
</year>
)
<title>
  Spinoza and Religion.
</title>
<container-title>
  The Philosophical Review
</container-title>,
<volume>
  16
</volume>
(
<issue>
  3
</issue>
), p.
<page>
  339
</page>
. [online] Available from:
<URL>
  http://dx.doi.org/10.2307/2177340
</URL>
```

Preprocesamiento CORA

En el caso del corpus CORA, las etiquetas fueron ajustadas para homologarlas con las utilizadas en el entrenamiento de Giant, adicionalmente se tuvieron que descartar 88 referencias por errores en su codificación y duplicidad de etiquetas, quedando para la evaluación 1787 de 1877 referencias.

En el caso del corpus CORA, las etiquetas se ajustaron para alinearse con las obtenidas con el preprocesamiento del corpus GIANT (ver tabla 3.1), lo que garantiza la coherencia en ambos conjuntos de datos. El mismo preproceso de ingeniería aplicada al corpus GIANT también se empleó para CORA, con el objetivo de estandarizar los datos y reducir la variabilidad y la complejidad. El preprocesamiento garantiza que ambos conjuntos de datos puedan funcionar correctamente para el entrenamiento y evaluación de los modelos, facilitando una comparación directa de su desempeño en datos bibliográficos estandarizados.

Tabla 3.1: Homologación de etiquetas CORA.

Etiqueta Cora	Nueva etiqueta
AUTHOR	AUTHOR
BOOKTITLE/JOURNAL	CONTAINER-TITLE
DATA	YEAR
PAGES	PAGES
PUBLISHER	PUBLISHER
VOLUME	VOLUME
TITLE	TITLE
TECH	<REMOVED>
INSTITUTE	<REMOVED>
EDITOR	<REMOVED>
NOTE	<REMOVED>

Preprocesamiento Redalyc

En el caso de los conjuntos Redalyc se construyeron referencias sintéticas a partir de la información Bibtex recuperada, generando 5000 referencias para el caso de español, 5000 referencias para el caso de inglés y 2000 referencias para el caso de francés, cada una con un estilo (APA6, MLA, Chicago, Vancouver, Harvard, ISO, NLM) elegido a asar.

3.5. Entorno de desarrollo

Para el desarrollo del modelo se utilizó el framework Flair NLP[9] creado por Akbik y cols. (2019) por las siguientes razones:

- Su capacidad para integrar y gestionar eficientemente diferentes tipos de incrustaciones.
- Arquitectura extensible y modular que facilita la adición de capas específicas del modelo, como Word Dropout y Locked Dropout.
- Documentación completa y ejemplos prácticos disponibles.

En adición al marco de NLP Flair, varias otras herramientas y bibliotecas fueron integrales en la implementación:

- PyTorch: Sirve como el marco subyacente para Flair, permitiendo grafos de computación dinámica y operaciones tensoriales para el entrenamiento y evaluación del modelo. PyTorch también se utilizó para desarrollar la arquitectura del codificador Transformer, permitiendo personalización para la segmentación de referencias bibliográficas.

[9] https://flairnlp.github.io/

- Pandas: Utilizado para la manipulación y análisis de datos, ayudando en la organización y formateo de datos antes del entrenamiento del modelo.

Requisitos de computación:

- Los modelos fueron entrenados y evaluados en una máquina equipada con un procesador Intel i9 y 64 GB de RAM, que soporta el procesamiento de grandes conjuntos de datos.
- Se empleó la aceleración por GPU para mejorar la velocidad de entrenamiento del modelo, utilizando una NVIDIA A5000.
- Se asignaron aproximadamente 1000 GB de almacenamiento SSD para almacenar tanto los conjuntos de datos en bruto como procesados, así como el estado de los modelos durante diferentes fases del entrenamiento.

3.6. Arquitecturas implementadas para experimentación

Como se ha mencionado previamente, en este trabajo se proponen siete modelos que emplean BPEmb como capa de selección de características y uno que utiliza Word2Vec. Para la fase de captura de contexto, se han adaptado varias arquitecturas reconocidas en el estado del arte.

El primer modelo es CRF, un enfoque sin captura explícita de contexto. Le sigue BiLSTM, la arquitectura más popular en tareas de secuencias, y Word2Vec + BiLSTM, un modelo que utiliza Word2Vec como capa de selección de características para ser el punto de comparación más cercano al estado del arte, manteniendo las mismas etiquetas y métodos de tokenización que los modelos propuestos con BPEmb.

Adicionalmente, se incluye el codificador Transformer, conocido por su capacidad para capturar dependencias a larga distancia dentro de las secuencias. Como parte de las contribuciones de este trabajo, se propone una arquitectura híbrida que combina BiLSTM y Transformer, aprovechando las fortalezas de ambos enfoques.

Finalmente, se introduce una modificación en la capa de atención del Transformer Encoder, pasando de una atención generalizada a una localizada mediante una técnica de ventana deslizante. De esta modificación surgen dos nuevas arquitecturas: Sliding Window y Sliding Window + BiLSTM, que buscan capturar patrones contextuales más específicos de manera eficiente.

Los modelos desarrollados a lo largo del estudio comparten una arquitectura base común, que emplea BPEmb en siete de ellos y Word2Vec en uno, concatenados con Character Embeddings para la representación vectorial, constituyendo así la fase de selección de características.

Además de las capas de representación, la arquitectura común de los modelos incluye varias capas adicionales para optimizar el rendimiento y la generalización:

- Word Dropout: Capa reduce el sobreajuste al apagar (es decir, poner a cero) de manera aleatoria, ciertos vectores de palabras durante el entrenamiento, lo que ayuda a que el modelo no dependa demasiado de palabras específicas.

- Locked Dropout: Similar al word dropout, pero se aplica de manera uniforme a todas las dimensiones de un vector de palabra en un paso dado. Esto mejora la robustez del modelo al evitar que se sobreajuste a patrones específicos en los datos de entrenamiento.
- Embedding2NN: Una capa que transforma los embeddings concatenados a una representación más adecuada para el procesamiento por las capas subsiguientes. Dicha transformación incluye operaciones no lineales para capturar relaciones más complejas en los datos.
- Linear: Una capa lineal que actúa como un clasificador, mapeando las representaciones procesadas a las etiquetas objetivo de segmentación de referencias.

Por último, todos los modelos culminan en una capa de predicción implementada con CRF.

Se hace notar que cada modelo incorpora una capa de procesamiento específica (captura de contexto) que aprovecha las fortalezas de cada enfoque.

3.6.1. Modelo CRF

El modelo CRF no cuenta con fase de captura de contexto, es decir, se centra únicamente en el uso de campos aleatorios condicionales para la segmentación de referencias. Diseñado para determinar el impacto de la falta de la fase de captura de contexto (ver figura 3.3).

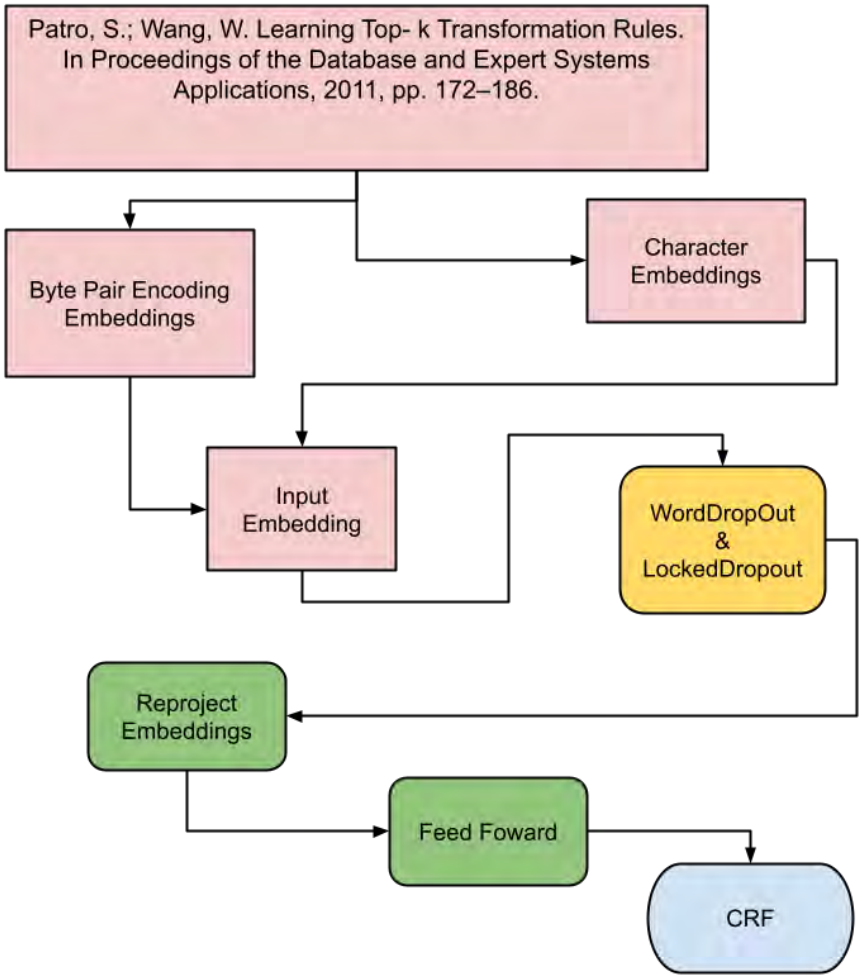


Figura 3.3: Representación gráfica del modelo CRF (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo CRF desarrollada con Flair NLP, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "CRF(
 #Input Embeddings
 (embeddings): StackedEmbeddings(
  (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)
  (list_embedding_1): CharacterEmbeddings(
   (char_embedding): Embedding(275, 25)
   (char_rnn): LSTM(25, 25, bidirectional=True)
```

```
  )
 )
 (word_dropout): WordDropout(p=0.05)
 (locked_dropout): LockedDropout(p=0.5)
 # Reproject Embedings
 (embedding2nn): Linear(in_features=650, out_features=650, bias=True)
 #Feed Foward
 (linear): Linear(in_features=650, out_features=29, bias=True)
 (loss_function): ViterbiLoss()
 (crf): CRF()
)"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo CRF:

Incrustaciones de Tokens y Caracteres: Esta ecuación representa la concatenación de dos tipos de incrustaciones (embeddings) para un token w_i:

$$\text{emb}(w_i) = [\text{emb}_{\text{BPE}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.1}$$

Donde:

- $\text{emb}_{\text{BPE}}(w_i)$: Incrustación basada en Byte Pair Encoding.
- $\text{emb}_{\text{char}}(w_i)$: Incrustación basada en caracteres.

Ambas se combinan para capturar tanto la semántica como la morfología del token.

Word Dropout: Este mecanismo aplica un apagado aleatorio sobre los embeddings para evitar sobreajuste:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{emb}(w_i), p = 0.05) \tag{3.2}$$

Esto significa que cada token tiene una probabilidad del 5 % de ser desactivado durante el entrenamiento.

Locked Dropout: Similar al Word Dropout, pero se aplica uniformemente a todas las dimensiones del vector de embedding:

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.3}$$

Con una probabilidad del 50 %, todo el vector del embedding es bloqueado.

Transformación Lineal (Embedding to NN): Esta capa lineal ajusta la dimensionalidad del embedding para preparar su procesamiento por capas posteriores:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.4}$$

Aquí, se mapea el embedding a un espacio de 650 dimensiones.

Capa Lineal para Clasificación: Después de procesar los embeddings, esta capa asigna las representaciones a una etiqueta de salida:

$$\text{output}(w_i) = \text{Linear}(\text{emb}_{\text{nn}}(w_i), 29) \tag{3.5}$$

Donde 29 es el número de posibles etiquetas para la segmentación de referencias.

CRF (Conditional Random Field): Esta ecuación define la probabilidad de una secuencia de etiquetas Y dada una secuencia de entradas H:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.6}$$

Donde:

- T_{y_{i-1},y_i}: Transición entre las etiquetas y_{i-1} y y_i.
- S_{i,y_i}: Puntuación del modelo para la etiqueta y_i en la posición i.

Finalmente, se busca la secuencia de etiquetas Y^* que maximiza esta probabilidad:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.7}$$

3.6.2. Modelo BiLSTM

El modelo BiLSTM combina redes neuronales recurrentes bidireccionales (BiLSTM) con CRF para una captura del contexto tanto pasado como futuro en la secuencia de tokens. Las BiLSTMs procesan la secuencia en ambas direcciones, ofreciendo una comprensión más profunda de la estructura sintáctica (ver figura 3.4).

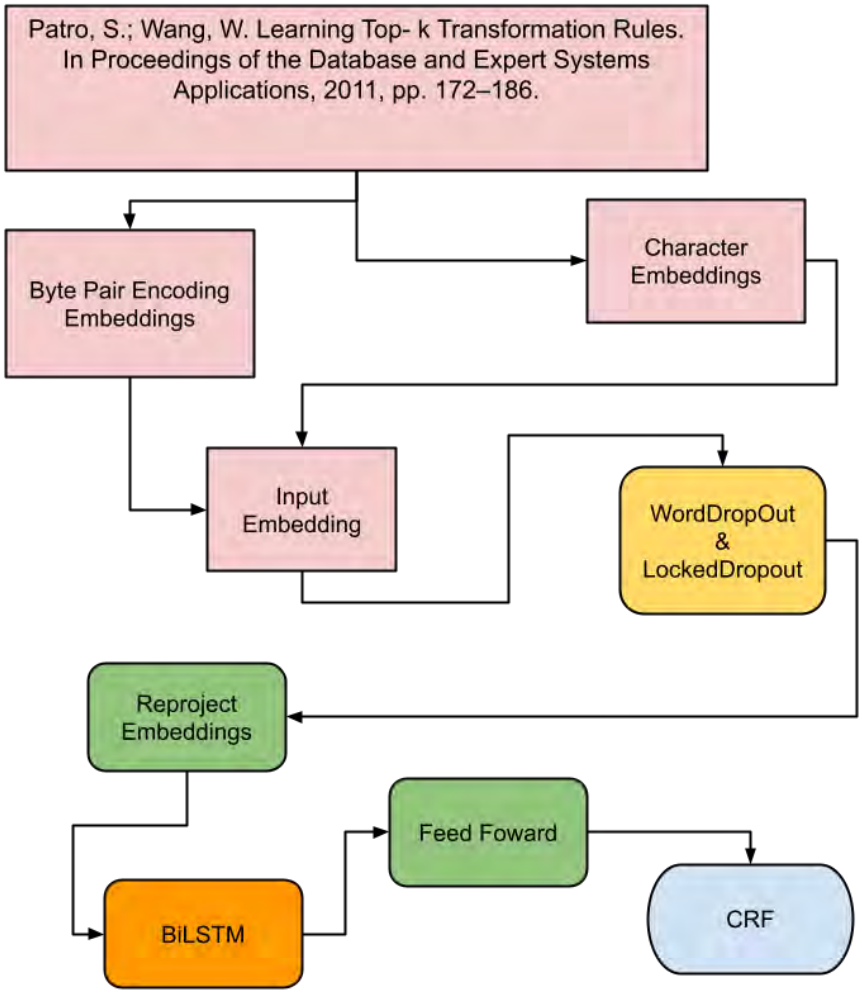


Figura 3.4: Representación gráfica del modelo BiLSTM (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo BiLSTM desarrollada con Flair NLP, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "BiLSTM(
 #Input Embedding
 (embeddings): StackedEmbeddings(
  (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)
  (list_embedding_1): CharacterEmbeddings(
   (char_embedding): Embedding(275, 25)
```

```
   (char_rnn): LSTM(25, 25, bidirectional=True)
  )
 )
 (word_dropout): WordDropout(p=0.05)
 (locked_dropout): LockedDropout(p=0.5)
 #Reproject Embeddings
 (embedding2nn): Linear(in_features=650, out_features=650, bias=True)
 #BiLSTM
 (rnn): LSTM(650, 256, batch\_first=True, bidirectional=True)
 #Feed Foward
 (linear): Linear(in_features=512, out_features=29, bias=True)
 (loss_function): ViterbiLoss()
 (crf): CRF()
)"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo BiLSTM:

Incrustaciones de Tokens y Caracteres: Esta ecuación combina las incrustaciones basadas en Byte Pair Encoding (BPE) y en caracteres para cada token w_i:

$$\text{emb}(w_i) = [\text{emb}_{\text{BPE}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.8}$$

Donde:

- $\text{emb}_{\text{BPE}}(w_i)$: Captura la información semántica basada en subpalabras mediante BPE.
- $\text{emb}_{\text{char}}(w_i)$: Captura información morfológica mediante caracteres individuales.

Word Dropout: Este mecanismo evita el sobreajuste durante el entrenamiento apagando aleatoriamente embeddings de palabras:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{emb}(w_i), p = 0.05) \tag{3.9}$$

Esto significa que cada token tiene una probabilidad del 5 % de ser omitido en cada iteración.

Locked Dropout: Este tipo de dropout se aplica uniformemente en todas las dimensiones de un embedding para mejorar la robustez:

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.10}$$

Con una probabilidad del 50 %, el vector completo es apagado de forma consistente durante una secuencia completa.

Transformación Lineal (Embedding to NN): Esta capa transforma los embeddings hacia un espacio de 650 dimensiones, preparándolos para el procesamiento por capas subsiguientes:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.11}$$

BiLSTM: El modelo BiLSTM captura la información contextual tanto hacia adelante como hacia atrás en la secuencia:

$$\overrightarrow{h_i} = \text{BiLSTM}_{\text{forward}}(\text{emb}_{\text{nn}}(w_i), \overrightarrow{h_{i-1}}) \tag{3.12}$$

$$\overleftarrow{h_i} = \text{BiLSTM}_{\text{backward}}(\text{emb}_{\text{nn}}(w_i), \overleftarrow{h_{i+1}}) \tag{3.13}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{3.14}$$

Donde:

- $\overrightarrow{h_i}$: Contexto hacia adelante.
- $\overleftarrow{h_i}$: Contexto hacia atrás.
- h_i: Concatenación de ambos contextos.

Capa Lineal para Clasificación: Después de procesar los embeddings con BiLSTM, esta capa asigna una etiqueta de salida:

$$\text{output}(w_i) = \text{Linear}(h_i, 29) \tag{3.15}$$

Donde 29 es el número de posibles etiquetas para la segmentación de referencias.

CRF (Conditional Random Field): La probabilidad de una secuencia de etiquetas Y dada una secuencia de entradas H se define como:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.16}$$

Donde:

- T_{y_{i-1},y_i}: Puntuación de transición entre las etiquetas y_{i-1} y y_i.
- S_{i,y_i}: Puntuación para la etiqueta y_i en la posición i.

Finalmente, se selecciona la secuencia Y^* que maximiza la probabilidad:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.17}$$

3.6.3. Modelo Word2Vec + BiLSTM

El modelo Word2Vec + BiLSTM se diseñó para ofrecer una comparación lo más cercana posible a la literatura, manteniendo condiciones uniformes con el resto de los modelos. Es idéntico al modelo BiLSTM (ver sección 3.6.2), su única diferencia es que utiliza el modelo Word2Vec preentrenado conocido como GoogleNews-vectors-negative300 para la capa de incrustaciones de palabras, que es el modelo más popular en el estado del arte para este propósito (ver figura 3.5).

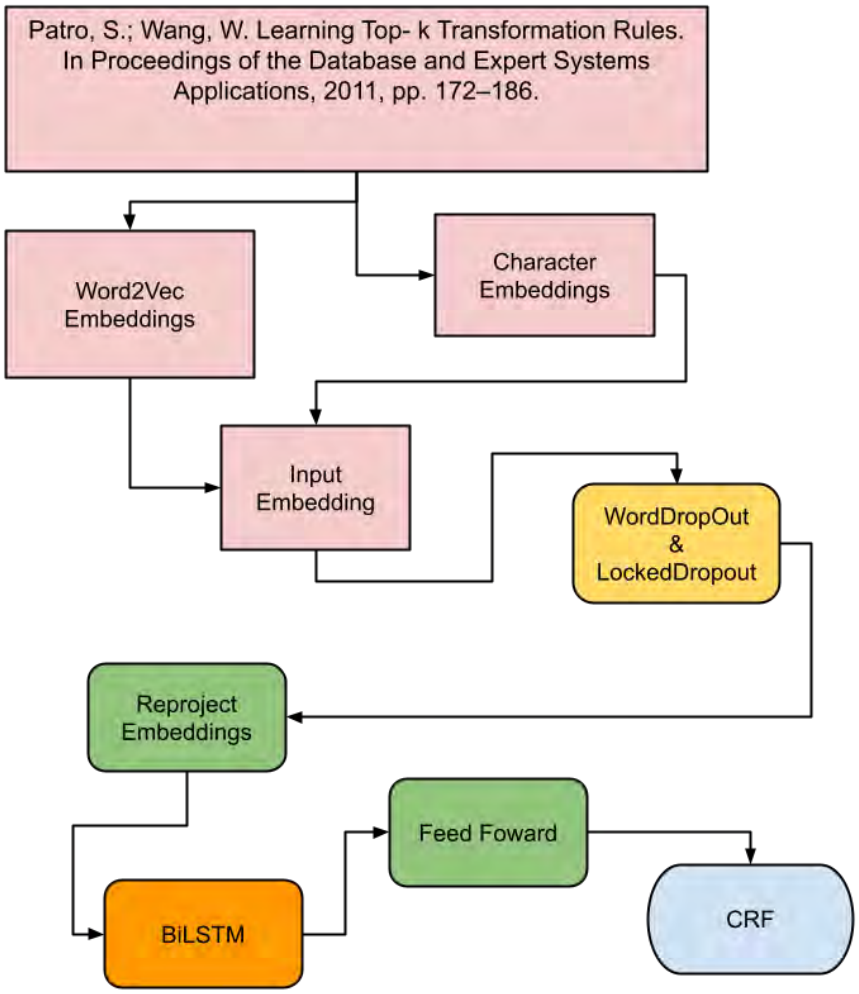


Figura 3.5: Representación gráfica del modelo Word2Vec + BiLSTM (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo Word2Vec + BiLSTM desarrollada con Flair NLP, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "BiLSTM + Word2Vec(
 #Input Embedding
 (embeddings): StackedEmbeddings(
  (list_embedding_0): WordEmbeddings(
   'GoogleNews-vectors-negative300.bin'
```

```
   (embedding): Embedding(3000001, 300)
  )
  (list_embedding_1): CharacterEmbeddings(
   (char_embedding): Embedding(275, 25)
   (char_rnn): LSTM(25, 25, bidirectional=True)
  )
 )
 (word_dropout): WordDropout(p=0.05)
 (locked_dropout): LockedDropout(p=0.5)
 #Reproject Embeddings
 (embedding2nn): Linear(in_features=650, out_features=650, bias=True)
 #BiLSTM
 (rnn): LSTM(650, 256, batch\_first=True, bidirectional=True)
 #Feed Foward
 (linear): Linear(in_features=512, out_features=29, bias=True)
 (loss_function): ViterbiLoss()
 (crf): CRF()
)"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo Word2Vec BiLSTM:

Incrustaciones de Tokens y Caracteres: Esta ecuación representa la combinación de dos tipos de embeddings: uno basado en palabras y otro basado en caracteres, para capturar tanto información semántica como morfológica:

$$\text{emb}(w_i) = [\text{emb}_{\text{Word}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.18}$$

Donde:

- $\text{emb}_{\text{Word}}(w_i)$: Embedding basado en la palabra completa.
- $\text{emb}_{\text{char}}(w_i)$: Embedding basado en caracteres individuales.

Word Dropout: Aplica un mecanismo de dropout para evitar que el modelo dependa demasiado de palabras específicas:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{emb}(w_i), p = 0.05) \tag{3.19}$$

Esto significa que cada embedding tiene un 5 % de probabilidad de ser desactivado durante el entrenamiento.

Locked Dropout: Un tipo de dropout que se aplica a todo el vector del embedding de forma consistente durante una secuencia completa:

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.20}$$

Esto garantiza que el modelo no dependa de ciertas dimensiones del embedding.

Transformación Lineal (Embedding to NN): Convierte los embeddings a un espacio de 650 dimensiones, preparándolos para las capas siguientes:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.21}$$

BiLSTM: Procesa la secuencia en ambas direcciones para capturar tanto el contexto pasado como el futuro:

$$\overrightarrow{h_i} = \text{BiLSTM}_{\text{forward}}(\text{emb}_{\text{nn}}(w_i), \overrightarrow{h_{i-1}}) \tag{3.22}$$

$$\overleftarrow{h_i} = \text{BiLSTM}_{\text{backward}}(\text{emb}_{\text{nn}}(w_i), \overleftarrow{h_{i+1}}) \tag{3.23}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{3.24}$$

Donde:

- $\overrightarrow{h_i}$: Estado oculto en la dirección hacia adelante.
- $\overleftarrow{h_i}$: Estado oculto en la dirección hacia atrás.
- h_i: Concatenación de los estados en ambas direcciones.

Capa Lineal para Clasificación: Esta capa toma la salida del BiLSTM y la convierte en una etiqueta de clasificación:

$$\text{output}(w_i) = \text{Linear}(h_i, 29) \tag{3.25}$$

Donde 29 es el número de etiquetas posibles para la segmentación de referencias.

CRF (Conditional Random Field): El CRF calcula la probabilidad de una secuencia de etiquetas Y dado un conjunto de entradas H:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.26}$$

Donde:

- T_{y_{i-1},y_i}: Puntuación de transición entre etiquetas consecutivas.
- S_{i,y_i}: Puntuación para la etiqueta y_i en la posición i.

Finalmente, se selecciona la secuencia de etiquetas que maximiza la probabilidad:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.27}$$

3.6.4. Modelo Transformer

El modelo Transformer utiliza el codificador propuesto por Vaswani y cols. (2017), un codificador de posiciones y un mecanismo de atención global para procesar secuencias de datos. Este enfoque permite capturar relaciones complejas y no lineales dentro de la secuencia, mejorando así la capacidad del modelo para comprender y analizar la estructura de los datos (ver figura 3.6).

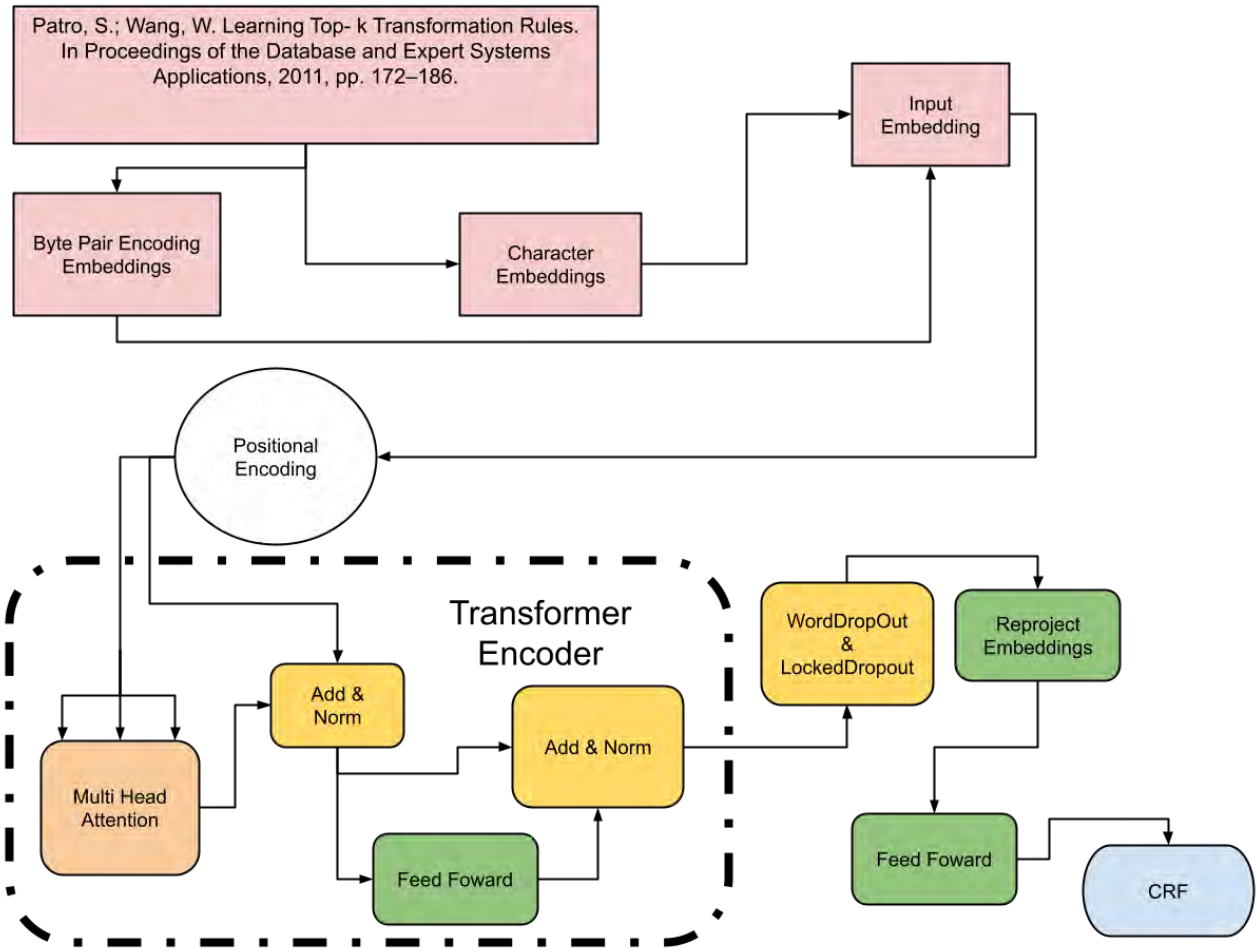


Figura 3.6: Representación gráfica del modelo Transformer (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo Transformer desarrollada con Flair NLP, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "Transformer(
 #Input Embedding
 (embeddings): CustomStackedEmbeddings(
  (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)
  (list_embedding_1): CharacterEmbeddings(
   (char_embedding): Embedding(275, 25)
```

```
  (char_rnn): LSTM(25, 25, bidirectional=True)
 )
 (positional_encoding): PositionalEncoding(
  (dropout): Dropout(p=0.1, inplace=False)
 )
 #Transformer Encoder
 (transformer_encoder_layer): CustomTransformerEncoderLayer(
  (self_attn): MultiheadAttention(
   (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
   out_features=256, bias=True)
  )
  (linear1): Linear(in_features=256, out_features=512, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (linear2): Linear(in_features=512, out_features=256, bias=True)
  (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (dropout1): Dropout(p=0.1, inplace=False)
  (dropout2): Dropout(p=0.1, inplace=False)
  (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
  track_running_stats=True)
 )
 (transformer_encoder): TransformerEncoder(
  (layers): ModuleList(
   (0): CustomTransformerEncoderLayer(
    (self_attn): MultiheadAttention(
     (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
     out_features=256, bias=True)
    )
    (linear1): Linear(in_features=256, out_features=512, bias=True)
    (dropout): Dropout(p=0.1, inplace=False)
    (linear2): Linear(in_features=512, out_features=256, bias=True)
    (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
    (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
    (dropout1): Dropout(p=0.1, inplace=False)
    (dropout2): Dropout(p=0.1, inplace=False)
    (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
    track_running_stats=True)
   )
  )
 )
)
(word_dropout): WordDropout(p=0.05)
(locked_dropout): LockedDropout(p=0.5)
#Reproject Embeddings
(embedding2nn): Linear(in_features=650, out_features=650, bias=True)
#Feed Foward
(linear): Linear(in_features=650, out_features=29, bias=True)
```

```
 (loss_function): ViterbiLoss()
 (crf): CRF()
)"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo Transformer:

Incrustaciones de Tokens y Caracteres: Esta ecuación combina las incrustaciones basadas en subpalabras mediante Byte Pair Encoding (BPE) y en caracteres, para capturar tanto la semántica como la morfología de los tokens:

$$\text{emb}(w_i) = [\text{emb}_{\text{BPE}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.28}$$

Codificación Posicional: Se aplica codificación posicional para incorporar información sobre la posición de cada token en la secuencia, lo cual es esencial para que el modelo Transformer capture las dependencias a lo largo de la secuencia:

$$\text{emb}_{\text{pos}}(w_i) = \text{PositionalEncoding}(\text{emb}(w_i)) \tag{3.29}$$

Encoder de Transformador: El encoder del Transformer se basa en mecanismos de atención multi-cabeza para capturar relaciones entre los tokens:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.30}$$

Donde:

- Q son las consultas (queries).
- K son las claves (keys).
- V son los valores (values).
- d_k es la dimensión de las claves.

El encoder aplica esta atención en varias capas para procesar los embeddings posicionales:

$$\text{transformer\_out}(w_i) = \text{TransformerEncoderLayer}(\text{emb}_{\text{pos}}(w_i)) \tag{3.31}$$

Las operaciones dentro de un encoder se detallan como sigue:

$$\text{X}_{\text{input}} = \text{emb}_{\text{pos}}(w_i) \tag{3.32}$$
$$\text{X}_{\text{att}} = \text{MultiheadAttention}(\text{X}_{\text{input}}, \text{X}_{\text{input}}, \text{X}_{\text{input}}) \tag{3.33}$$
$$\text{X}_{\text{dropout}} = \text{Dropout}(\text{X}_{\text{att}}) + \text{X}_{\text{input}} \tag{3.34}$$
$$\text{X}_{\text{norm1}} = \text{LayerNorm}(\text{X}_{\text{dropout}}) \tag{3.35}$$
$$\text{X}_{\text{intermediate}} = \text{Linear1}(\text{X}_{\text{norm1}}) \tag{3.36}$$
$$\text{X}_{\text{output}} = \text{Linear2}(\text{X}_{\text{intermediate}}) \tag{3.37}$$
$$\text{X}_{\text{dropout2}} = \text{Dropout}(\text{X}_{\text{output}}) + \text{X}_{\text{norm1}} \tag{3.38}$$
$$\text{transformer\_out}(w_i) = \text{LayerNorm}(\text{X}_{\text{dropout2}}) \tag{3.39}$$

Word Dropout y Locked Dropout: Para evitar sobreajuste, se aplican dos tipos de dropout:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{transformer\_out}(w_i), p = 0.05) \tag{3.40}$$

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.41}$$

El Locked Dropout mantiene la coherencia del apagado a lo largo de la secuencia.

Transformación Lineal (Embedding to NN): Convierte los embeddings a un espacio de 650 dimensiones:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.42}$$

Capa Lineal para Clasificación: Esta capa asigna etiquetas a cada token:

$$\text{output}(w_i) = \text{Linear}(\text{emb}_{\text{nn}}(w_i), 29) \tag{3.43}$$

Donde 29 es el número de etiquetas posibles.

CRF (Conditional Random Field): El CRF calcula la probabilidad de una secuencia de etiquetas:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.44}$$

Donde:

- T_{y_{i-1},y_i}: Puntuación de transición entre etiquetas.
- S_{i,y_i}: Puntuación de la etiqueta en la posición i.

Finalmente, se selecciona la secuencia de etiquetas más probable:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.45}$$

Se debe mencionar que en el caso del modelo Transformer, la colocación de la capa Transformer Encoder antes de la capa embedding2nn obedece las siguientes razones.

Primero, en las referencias bibliográficas, el contexto es de gran relevancia. La posición de una palabra o frase puede alterar significativamente su interpretación, como por ejemplo, distinguir entre el nombre de un autor y el título de un trabajo. Al procesar las incrustaciones a través del codificador posicional y el Transformer desde el inicio, el modelo puede capturar de manera más efectiva las relaciones contextuales y estructurales específicas de las referencias bibliográficas.

Segundo, la inclusión temprana del Transformer permite una captura precoz de las relaciones contextuales. Lo que puede ser determinante en las referencias bibliográficas, donde la estructura y el orden de los elementos (autores, título, año de publicación, etc.) siguen patrones que pueden ser complejos y variados. El Transformer, conocido por su habilidad para manejar dependencias a larga distancia, es teóricamente significativo para detectar y aprender patrones.

Finalmente, tras generar las representaciones contextualizadas, la capa embedding2nn funciona como un mecanismo de ajuste fino, reduciendo la complejidad computacional y optimizando cada elemento. REfinando los datos resultantes, haciéndolos aún más adecuados para la identificación precisa de los diversos componentes dentro de las referencias bibliográficas.

3.6.5. Modelo Sliding Window

El modelo Sliding Window se diseñó con el propósito de adaptar el funcionamiento de los mecanismos de atención a la naturaleza sintáctica de las referencias bibliográficas. Es una copia casi exacta del modelo Transformer (Sección 3.6.4), solamente difieren en la forma de trabajo del mecanismo de atención. A continuación se explican las diferencias:

El modelo Transformer aplica una atención estándar como propone el modelo de Vaswani y cols. (2017), es decir, obtener el producto escalar seguido de una función softmax, que se representa con la siguiente ecuación (5.46):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.46}$$

donde Q son las consultas, K son las claves, V son los valores, y d_k es la dimensión de las claves.

En cambio el modelo Sliding Window aplica una función de atención personalizada que modifica la forma en que se calculan las puntuaciones de atención. En lugar de usar directamente el producto escalar seguido de una función softmax, se introduce una máscara de ventana deslizante (window_mask) y ajustes para manejar tokens de puntuación, que se representa con la siguiente ecuación:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \odot \text{window\_mask}\right)V \tag{3.47}$$

$$\text{window\_mask}_{ij} = \begin{cases} 1 & \text{si } |i-j| \leq \text{window\_size o is\_punctuation}(i) \text{ o is\_punctuation}(j) \\ 0 & \text{en otro caso} \end{cases} \tag{3.48}$$

- Q son las consultas.
- K son las claves.
- V son los valores.
- d_k es la dimensión de las representaciones de las consultas y las claves.
- window_mask es la máscara de ventana deslizante.

donde:

- $W_{i,j}$ es el peso de atención calculado como el producto punto entre la consulta Q_i y la clave K_j.
- $Q_{i,k}$ y $K_{j,k}$ son las componentes de la consulta Q_i y la clave K_j en la posición k, respectivamente.
- d_k es la dimensión de las representaciones de consultas y claves.
- window_mask_{ij} es la máscara de ventana deslizante aplicada entre la consulta Q_i y la clave K_j.
- $|i-j|$ es la distancia relativa entre la consulta Q_i y la clave K_j.

- window_size es el tamaño de la ventana deslizante.
- is_punctuation(i) y is_punctuation(j) son funciones que determinan si las posiciones i o j son tokens de puntuación.
- $\odot$ denota el producto elemento a elemento.
- $\sum$ masked_attention_scores es la suma de los scores de atención en la dimensión correspondiente para renormalizar.
- $\times$ denota la multiplicación de matrices.

En otras palabras, la máscara de atención considera los tokens que comparten la misma etiqueta, agrupándolos y asegurando que se les preste atención de manera conjunta. Esto se logra aplicando una máscara que forza la atención en los tokens que pertenecen a la misma clase o etiqueta, independientemente de su posición relativa. Para los tokens de puntuación, la máscara de atención también asegura que sean siempre considerados.

Esto significa que, incluso si un token de puntuación está dentro de un grupo de tokens, de la misma clase, la ventana deslizante lo identifica correctamente. De forma que el modelo puede mantener una comprensión del contexto y la estructura de la secuencia de entrada, capturando tanto las relaciones entre tokens etiquetados de manera similar como la información estructural proporcionada por los tokens de puntuación.

En la Figura 3.7 se puede apreciar una representación gráfica del modelo Sliding Window.

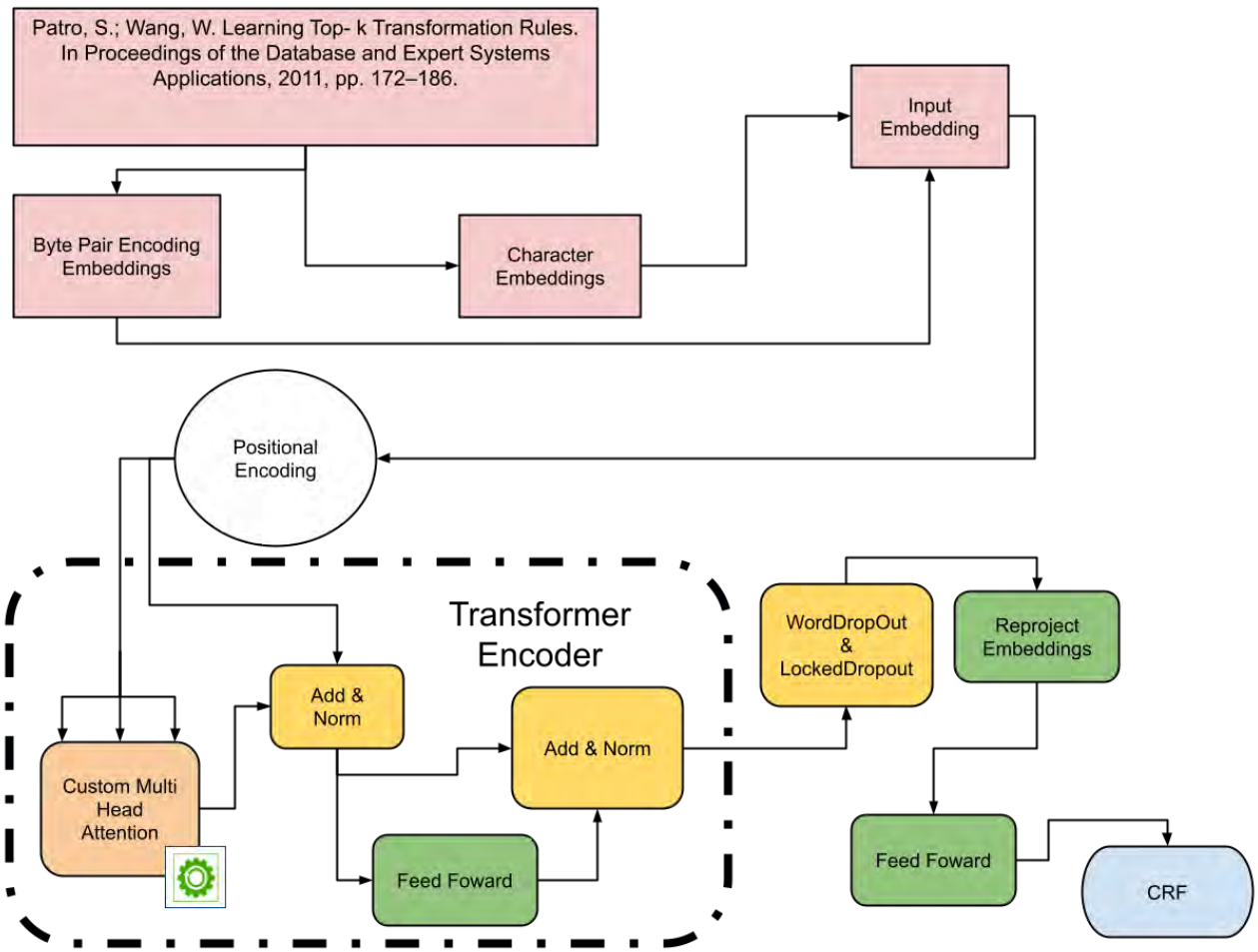


Figura 3.7: Representación gráfica del modelo Sliding Window (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo Sliding Window desarrollada con Flair NLP, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "Sliding Window(
#Input Embedding
(embeddings): CustomStackedEmbeddings(
 (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)
 (list_embedding_1): CharacterEmbeddings(
  (char_embedding): Embedding(275, 25)
  (char_rnn): LSTM(25, 25, bidirectional=True)
 )
 (positional_encoding): PositionalEncoding(
  (dropout): Dropout(p=0.1, inplace=False)
 )
 #Transformer Encoder - Sliding Window
 (transformer_encoder_layer): CustomTransformerEncoderLayer(
```

```
   #Sliding Window
   (self_attn): CustomAttention(
    (dropout): Dropout(p=0.1, inplace=False)
    (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
    out_features=256, bias=True)
   )
   (linear1): Linear(in_features=256, out_features=512, bias=True)
   (dropout): Dropout(p=0.1, inplace=False)
   (linear2): Linear(in_features=512, out_features=256, bias=True)
   (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
   (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
   (dropout1): Dropout(p=0.1, inplace=False)
   (dropout2): Dropout(p=0.1, inplace=False)
   (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
   track_running_stats=True)
  )
  (transformer_encoder): TransformerEncoder(
   (layers): ModuleList(
    (0): CustomTransformerEncoderLayer(
     (self_attn): CustomAttention(
      (dropout): Dropout(p=0.1, inplace=False)
      (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
      out_features=256, bias=True)
     )
     (linear1): Linear(in_features=256, out_features=512, bias=True)
     (dropout): Dropout(p=0.1, inplace=False)
     (linear2): Linear(in_features=512, out_features=256, bias=True)
     (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
     (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
     (dropout1): Dropout(p=0.1, inplace=False)
     (dropout2): Dropout(p=0.1, inplace=False)
     (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
     track_running_stats=True)
    )
   )
  )
 )
 (word_dropout): WordDropout(p=0.05)
 (locked_dropout): LockedDropout(p=0.5)
 #Reproject Embeddings
 (embedding2nn): Linear(in_features=650, out_features=650, bias=True)
 #Feed Foward
 (linear): Linear(in_features=650, out_features=29, bias=True)
 (loss_function): ViterbiLoss()
 (crf): CRF()
)"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo Sliding Window:

Incrustaciones de Tokens y Caracteres: Esta ecuación combina las incrustaciones basadas en Byte Pair Encoding (BPE) y en caracteres, capturando tanto las relaciones semánticas como las morfológicas de los tokens:

$$\text{emb}(w_i) = [\text{emb}_{\text{BPE}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.49}$$

Codificación Posicional: Se utiliza codificación posicional para incluir información sobre la posición de los tokens en la secuencia, permitiendo al Transformer capturar dependencias contextuales:

$$\text{emb}_{\text{pos}}(w_i) = \text{PositionalEncoding}(\text{emb}(w_i)) \tag{3.50}$$

Encoder de Transformador: El mecanismo de atención multi-cabeza del Transformer se representa por la siguiente ecuación, permitiendo capturar relaciones entre los tokens:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.51}$$

Donde:

- Q: Consultas (queries).
- K: Claves (keys).
- V: Valores (values).
- d_k: Dimensión de las claves.

El resultado del encoder del Transformer se calcula de la siguiente forma:

$$\text{transformer\_out}(w_i) = \text{TransformerEncoderLayer}(\text{emb}_{\text{pos}}(w_i)) \tag{3.52}$$

Detalles del Transformer Encoder Layer: Cada capa del encoder del Transformer aplica las siguientes operaciones secuenciales:

$$\text{X}_{\text{input}} = \text{emb}_{\text{pos}}(w_i) \tag{3.53}$$
$$\text{X}_{\text{att}} = \text{CustomAttention}(\text{X}_{\text{input}}, \text{X}_{\text{input}}, \text{X}_{\text{input}}) \tag{3.54}$$
$$\text{X}_{\text{dropout}} = \text{Dropout}(\text{X}_{\text{att}}) + \text{X}_{\text{input}} \tag{3.55}$$
$$\text{X}_{\text{norm1}} = \text{LayerNorm}(\text{X}_{\text{dropout}}) \tag{3.56}$$
$$\text{X}_{\text{intermediate}} = \text{Linear1}(\text{X}_{\text{norm1}}) \tag{3.57}$$
$$\text{X}_{\text{output}} = \text{Linear2}(\text{X}_{\text{intermediate}}) \tag{3.58}$$
$$\text{X}_{\text{dropout2}} = \text{Dropout}(\text{X}_{\text{output}}) + \text{X}_{\text{norm1}} \tag{3.59}$$
$$\text{transformer\_out}(w_i) = \text{LayerNorm}(\text{X}_{\text{dropout2}}) \tag{3.60}$$

Word Dropout y Locked Dropout: Se aplican dropout para evitar sobreajuste:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{transformer\_out}(w_i), p = 0.05) \tag{3.61}$$

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.62}$$

El Locked Dropout asegura que la desactivación sea coherente a lo largo de la secuencia.

Transformación Lineal (Embedding to NN): Convierte los embeddings al espacio requerido para la siguiente capa:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.63}$$

Capa Lineal para Clasificación: Asigna una etiqueta a cada token utilizando la salida procesada:

$$\text{output}(w_i) = \text{Linear}(\text{emb}_{\text{nn}}(w_i), 29) \tag{3.64}$$

Donde 29 es el número de etiquetas posibles.

CRF (Conditional Random Field): El modelo CRF calcula la probabilidad de una secuencia de etiquetas:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.65}$$

Donde:

- T_{y_{i-1},y_i}: Puntuación de transición entre etiquetas.
- S_{i,y_i}: Puntuación para la etiqueta en la posición i.

Finalmente, se selecciona la secuencia con la mayor probabilidad:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.66}$$

3.6.6. Modelo Transformer + BiLSTM

El modelo Transformer + BiLSTM es una copia exacta del modelo Transformer (ver sección 3.6.4), su única diferencia es que integra una capa adicional BiLSTM, justo después de la capa Reproject Embeddings (embedding2nn).

En la Figura 3.8 se puede observar una representación gráfica del modelo Transformer + BiLSTM:

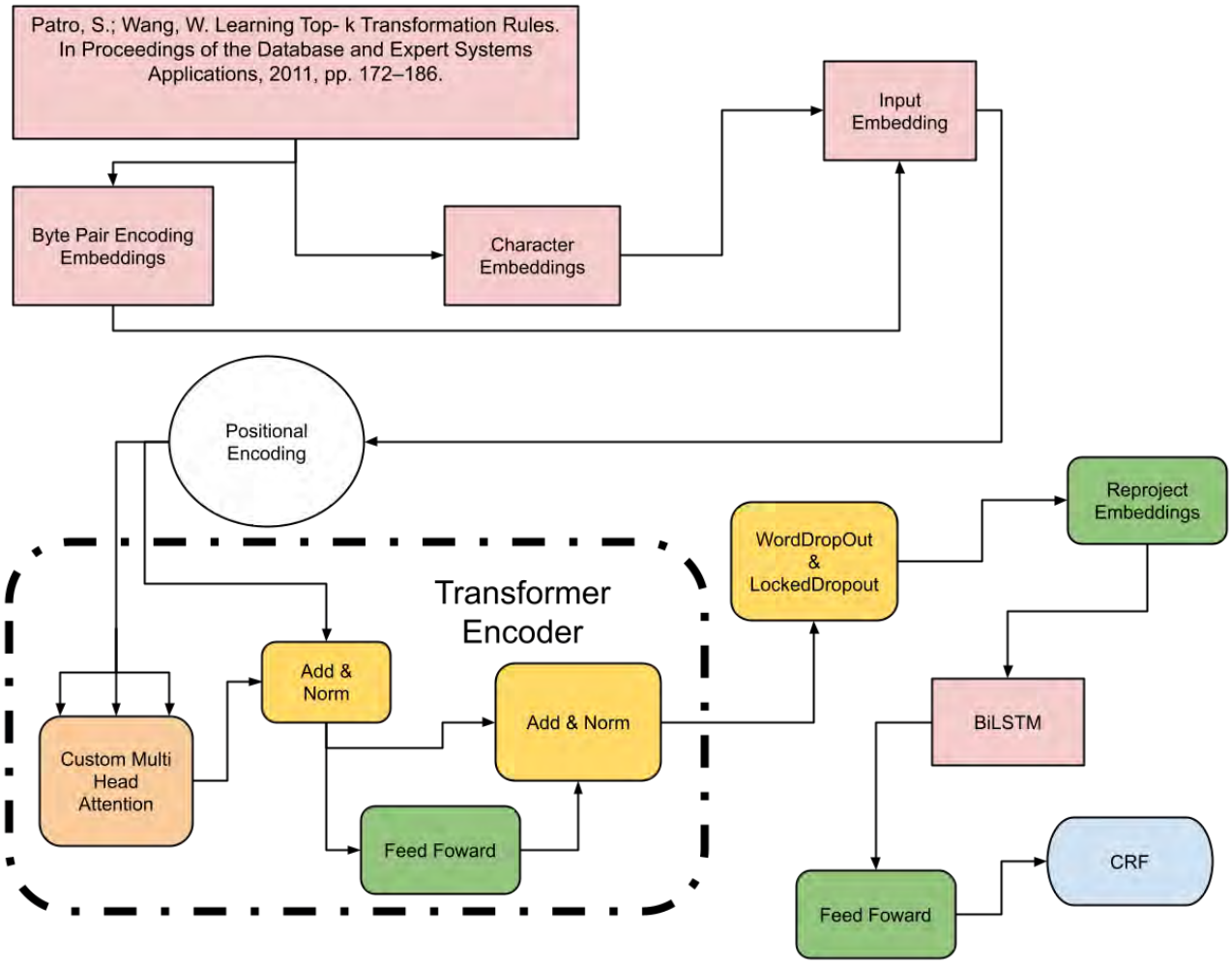


Figura 3.8: Representación gráfica del modelo Transformer + BiLSTM (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo Transformer + BiLSTM desarrollada con Flair NLP, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "Transformer(
 #Input Embedding
 (embeddings): CustomStackedEmbeddings(
  (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)
  (list_embedding_1): CharacterEmbeddings(
   (char_embedding): Embedding(275, 25)
```

```
  (char_rnn): LSTM(25, 25, bidirectional=True)
 )
 (positional_encoding): PositionalEncoding(
  (dropout): Dropout(p=0.1, inplace=False)
 )
 #Transformer Encoder
 (transformer_encoder_layer): CustomTransformerEncoderLayer(
  (self_attn): MultiheadAttention(
   (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
   out_features=256, bias=True)
  )
  (linear1): Linear(in_features=256, out_features=512, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (linear2): Linear(in_features=512, out_features=256, bias=True)
  (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (dropout1): Dropout(p=0.1, inplace=False)
  (dropout2): Dropout(p=0.1, inplace=False)
  (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
  track_running_stats=True)
 )
 (transformer_encoder): TransformerEncoder(
  (layers): ModuleList(
   (0): CustomTransformerEncoderLayer(
    (self_attn): MultiheadAttention(
     (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
     out_features=256, bias=True)
    )
    (linear1): Linear(in_features=256, out_features=512, bias=True)
    (dropout): Dropout(p=0.1, inplace=False)
    (linear2): Linear(in_features=512, out_features=256, bias=True)
    (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
    (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
    (dropout1): Dropout(p=0.1, inplace=False)
    (dropout2): Dropout(p=0.1, inplace=False)
    (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
    track_running_stats=True)
   )
  )
 )
)
(word_dropout): WordDropout(p=0.05)
(locked_dropout): LockedDropout(p=0.5)
#Reproject Embeddings
(embedding2nn): Linear(in_features=650, out_features=650, bias=True)
#BiLSTM
(rnn): LSTM(650, 256, batch\_first=True, bidirectional=True)
```

```
  #Feed Foward
  (linear): Linear(in_features=650, out_features=29, bias=True)
  (loss_function): ViterbiLoss()
  (crf): CRF()
 )"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo Transformer:

Incrustaciones de Tokens y Caracteres: Esta ecuación combina dos tipos de embeddings para cada token:

$$\text{emb}(w_i) = [\text{emb}_{\text{BPE}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.67}$$

- $\text{emb}_{\text{BPE}}(w_i)$: Captura relaciones semánticas mediante subpalabras.
- $\text{emb}_{\text{char}}(w_i)$: Captura morfología mediante caracteres.

Codificación Posicional: Se aplica para conservar la información del orden de los tokens en la secuencia:

$$\text{emb}_{\text{pos}}(w_i) = \text{PositionalEncoding}(\text{emb}(w_i)) \tag{3.68}$$

Encoder de Transformador: El encoder utiliza mecanismos de atención para aprender las relaciones contextuales:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.69}$$

Donde:

- Q: Consultas.
- K: Claves.
- V: Valores.
- d_k: Dimensión de las claves.

La salida del encoder se calcula como:

$$\text{transformer\_out}(w_i) = \text{TransformerEncoderLayer}(\text{emb}_{\text{pos}}(w_i)) \tag{3.70}$$

Detalles del Transformer Encoder Layer: Este bloque realiza las siguientes operaciones en cada capa:

$$\text{X}_{\text{input}} = \text{emb}_{\text{pos}}(w_i) \tag{3.71}$$

$$\text{X}_{\text{att}} = \text{MultiheadAttention}(\text{X}_{\text{input}}, \text{X}_{\text{input}}, \text{X}_{\text{input}}) \tag{3.72}$$

$$\text{X}_{\text{dropout}} = \text{Dropout}(\text{X}_{\text{att}}) + \text{X}_{\text{input}} \tag{3.73}$$

$$\text{X}_{\text{norm1}} = \text{LayerNorm}(\text{X}_{\text{dropout}}) \tag{3.74}$$

$$\text{X}_{\text{intermediate}} = \text{Linear1}(\text{X}_{\text{norm1}}) \tag{3.75}$$

$$\text{X}_{\text{output}} = \text{Linear2}(\text{X}_{\text{intermediate}}) \tag{3.76}$$

$$\text{X}_{\text{dropout2}} = \text{Dropout}(\text{X}_{\text{output}}) + \text{X}_{\text{norm1}} \tag{3.77}$$

$$\text{transformer\_out}(w_i) = \text{LayerNorm}(\text{X}_{\text{dropout2}}) \tag{3.78}$$

Word Dropout y Locked Dropout: Para evitar sobreajuste, se aplican estas técnicas:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{transformer\_out}(w_i), p = 0.05) \tag{3.79}$$

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.80}$$

Transformación Lineal (Embedding to NN): Convierte los embeddings a un espacio con 650 dimensiones:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.81}$$

BiLSTM: El modelo BiLSTM captura el contexto tanto hacia adelante como hacia atrás:

$$\overrightarrow{h_i} = \text{BiLSTM}_{\text{forward}}(\text{emb}_{\text{nn}}(w_i), \overrightarrow{h_{i-1}}) \tag{3.82}$$

$$\overleftarrow{h_i} = \text{BiLSTM}_{\text{backward}}(\text{emb}_{\text{nn}}(w_i), \overleftarrow{h_{i+1}}) \tag{3.83}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{3.84}$$

Capa Lineal para Clasificación: Esta capa asigna una etiqueta a cada token:

$$\text{output}(w_i) = \text{Linear}(\text{emb}_{\text{nn}}(w_i), 29) \tag{3.85}$$

CRF (Conditional Random Field): El CRF calcula la probabilidad de una secuencia de etiquetas:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.86}$$

Donde:

- T_{y_{i-1},y_i}: Puntuación de transición entre etiquetas consecutivas.
- S_{i,y_i}: Puntuación para la etiqueta en la posición i.

Finalmente, se selecciona la secuencia con mayor probabilidad:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.87}$$

El modelo Transformer + BiLSTM combina dos mecanismos de captura de contexto (Codificador de transformador y BiLSTM) para incrementar la eficiencia en la segmentación de referencias.

3.6.7. Modelo Sliding Window + BiLSTM

El modelo Sliding Window + BiLSTM es una copia exacta del modelo Sliding Window (ver sección 3.6.5), su única diferencia es que integra una capa adicional BiLSTM, justo después de la capa Reproject Embeddings (embedding2nn).

En la Figura 3.9 se puede observar una representación gráfica del modelo Transformer + BiLSTM:

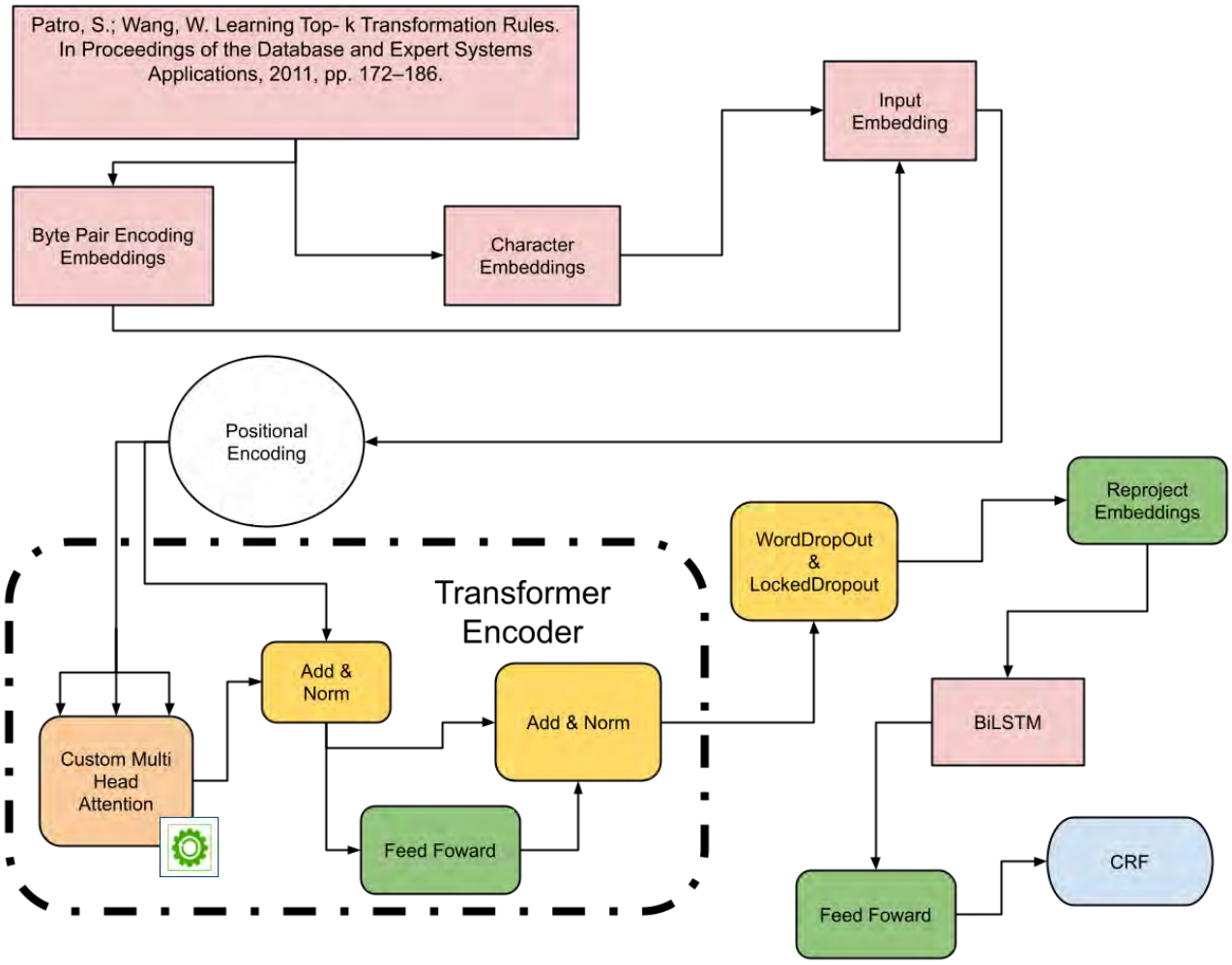


Figura 3.9: Representación gráfica del modelo Sliding Window + BiLSTM (los textos se presentan en inglés para mantener la coherencia con la nomenclatura utilizada en las capas de la arquitectura del modelo).

A continuación, se presenta la representación de la arquitectura del modelo Sliding Window + BiLSTM desarrollada con Flair sobre PyTorch, permitiendo su replicación por usuarios en esta herramienta.

```
Model: "Sliding Window + BiLSTM(
  #Input Embedding
 (embeddings): CustomStackedEmbeddings(
  (list_embedding_0): BytePairEmbeddings(model=0-bpe-multi-100000-50)
  (list_embedding_1): CharacterEmbeddings(
```

```
  (char_embedding): Embedding(275, 25)
  (char_rnn): LSTM(25, 25, bidirectional=True)
 )
 (positional_encoding): PositionalEncoding(
  (dropout): Dropout(p=0.1, inplace=False)
 )
 #Transformer Encoder
 (transformer_encoder_layer): CustomTransformerEncoderLayer(
  #Sliding Window
  (self_attn): CustomAttention(
   (dropout): Dropout(p=0.1, inplace=False)
   (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
   out_features=256, bias=True)
  )
  (linear1): Linear(in_features=256, out_features=512, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (linear2): Linear(in_features=512, out_features=256, bias=True)
  (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
  (dropout1): Dropout(p=0.1, inplace=False)
  (dropout2): Dropout(p=0.1, inplace=False)
  (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
  track_running_stats=True)
 )
 (transformer_encoder): TransformerEncoder(
  (layers): ModuleList(
   (0): CustomTransformerEncoderLayer(
    (self_attn): CustomAttention(
     (dropout): Dropout(p=0.1, inplace=False)
     (out_proj): NonDynamicallyQuantizableLinear(in_features=256,
     out_features=256, bias=True)
    )
    (linear1): Linear(in_features=256, out_features=512, bias=True)
    (dropout): Dropout(p=0.1, inplace=False)
    (linear2): Linear(in_features=512, out_features=256, bias=True)
    (norm1): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
    (norm2): LayerNorm((256,), eps=1e-06, elementwise_affine=True)
    (dropout1): Dropout(p=0.1, inplace=False)
    (dropout2): Dropout(p=0.1, inplace=False)
    (bn): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
    track_running_stats=True)
   )
  )
 )
)
(word_dropout): WordDropout(p=0.05)
(locked_dropout): LockedDropout(p=0.5)
```

```
 #Reproject Embeddings
 (embedding2nn): Linear(in_features=650, out_features=650, bias=True)
 #BiLSTM
 (rnn): LSTM(650, 256, batch\_first=True, bidirectional=True)
 #Feed Foward
 (linear): Linear(in_features=650, out_features=29, bias=True)
 (loss_function): ViterbiLoss()
 (crf): CRF()
)"
```

A continuación se pueden observar las ecuaciones que describen las interacciones de los componentes del modelo Sliding Window + BiLSTM:

Incrustaciones de Tokens y Caracteres: Esta ecuación combina las representaciones basadas en subpalabras (BPE) y caracteres para capturar tanto la semántica como la morfología del texto:

$$\text{emb}(w_i) = [\text{emb}_{\text{BPE}}(w_i); \text{emb}_{\text{char}}(w_i)] \tag{3.88}$$

Codificación Posicional: Incorpora la información de la posición de los tokens en la secuencia para que el Transformer capture dependencias contextuales:

$$\text{emb}_{\text{pos}}(w_i) = \text{PositionalEncoding}(\text{emb}(w_i)) \tag{3.89}$$

Encoder de Transformador: El mecanismo de atención multi-cabeza permite que el Transformer aprenda relaciones contextuales complejas:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.90}$$

Donde:

- Q: Consultas (queries).
- K: Claves (keys).
- V: Valores (values).
- d_k: Dimensión de las claves.

La salida del encoder se calcula como:

$$\text{transformer\_out}(w_i) = \text{TransformerEncoderLayer}(\text{emb}_{\text{pos}}(w_i)) \tag{3.91}$$

Detalles del Transformer Encoder Layer: Cada capa del encoder aplica estas operaciones:

$$\text{X}_{\text{input}} = \text{emb}_{\text{pos}}(w_i) \tag{3.92}$$

$$\text{X}_{\text{att}} = \text{CustomAttention}(\text{X}_{\text{input}}, \text{X}_{\text{input}}, \text{X}_{\text{input}}) \tag{3.93}$$

$$\text{X}_{\text{dropout}} = \text{Dropout}(\text{X}_{\text{att}}) + \text{X}_{\text{input}} \tag{3.94}$$

$$\text{X}_{\text{norm1}} = \text{LayerNorm}(\text{X}_{\text{dropout}}) \tag{3.95}$$

$$\text{X}_{\text{intermediate}} = \text{Linear1}(\text{X}_{\text{norm1}}) \tag{3.96}$$

$$\text{X}_{\text{output}} = \text{Linear2}(\text{X}_{\text{intermediate}}) \tag{3.97}$$

$$\text{X}_{\text{dropout2}} = \text{Dropout}(\text{X}_{\text{output}}) + \text{X}_{\text{norm1}} \tag{3.98}$$

$$\text{transformer\_out}(w_i) = \text{LayerNorm}(\text{X}_{\text{dropout2}}) \tag{3.99}$$

Word Dropout y Locked Dropout: Estas técnicas evitan sobreajuste:

$$\text{emb}_{\text{dropout}}(w_i) = \text{Dropout}(\text{transformer\_out}(w_i), p = 0.05) \tag{3.100}$$

$$\text{emb}_{\text{locked}}(w_i) = \text{LockedDropout}(\text{emb}_{\text{dropout}}(w_i), p = 0.5) \tag{3.101}$$

Transformación Lineal (Embedding to NN): Convierte los embeddings al espacio adecuado para la capa siguiente:

$$\text{emb}_{\text{nn}}(w_i) = \text{Linear}(\text{emb}_{\text{locked}}(w_i), 650) \tag{3.102}$$

BiLSTM: El BiLSTM captura el contexto tanto hacia adelante como hacia atrás en la secuencia:

$$\overrightarrow{h_i} = \text{BiLSTM}_{\text{forward}}(\text{emb}_{\text{nn}}(w_i), \overrightarrow{h_{i-1}}) \tag{3.103}$$

$$\overleftarrow{h_i} = \text{BiLSTM}_{\text{backward}}(\text{emb}_{\text{nn}}(w_i), \overleftarrow{h_{i+1}}) \tag{3.104}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{3.105}$$

Capa Lineal para Clasificación: Asigna una etiqueta a cada token:

$$\text{output}(w_i) = \text{Linear}(\text{emb}_{\text{nn}}(w_i), 29) \tag{3.106}$$

Donde 29 es el número de etiquetas posibles.

CRF (Conditional Random Field): El CRF calcula la probabilidad de una secuencia de etiquetas:

$$P(Y|H) = \frac{\exp(\sum_{i=1}^{n} T_{y_{i-1},y_i} + \sum_{i=1}^{n} S_{i,y_i})}{\sum_{Y'} \exp(\sum_{i=1}^{n} T_{y'_{i-1},y'_i} + \sum_{i=1}^{n} S_{i,y'_i})} \tag{3.107}$$

Donde:

- T_{y_{i-1},y_i}: Puntuación de transición entre etiquetas consecutivas.
- S_{i,y_i}: Puntuación de la etiqueta en la posición i.

Finalmente, se selecciona la secuencia con la mayor probabilidad:

$$Y^* = \arg\max_{Y} P(Y|H) \tag{3.108}$$

El modelo Sliding Window + BiLSTM combina dos mecanismos de captura de contexto (Codificador de transformador con atención personalizada y BiLSTM) para incrementar la eficiencia en la segmentación de referencias.

3.6.8. Complejidad computacional

La complejidad computacional, de los modelos descritos en este capítulo, se puede analizar en función del número de parámetros que cada uno posee, ver Figura 3.10.

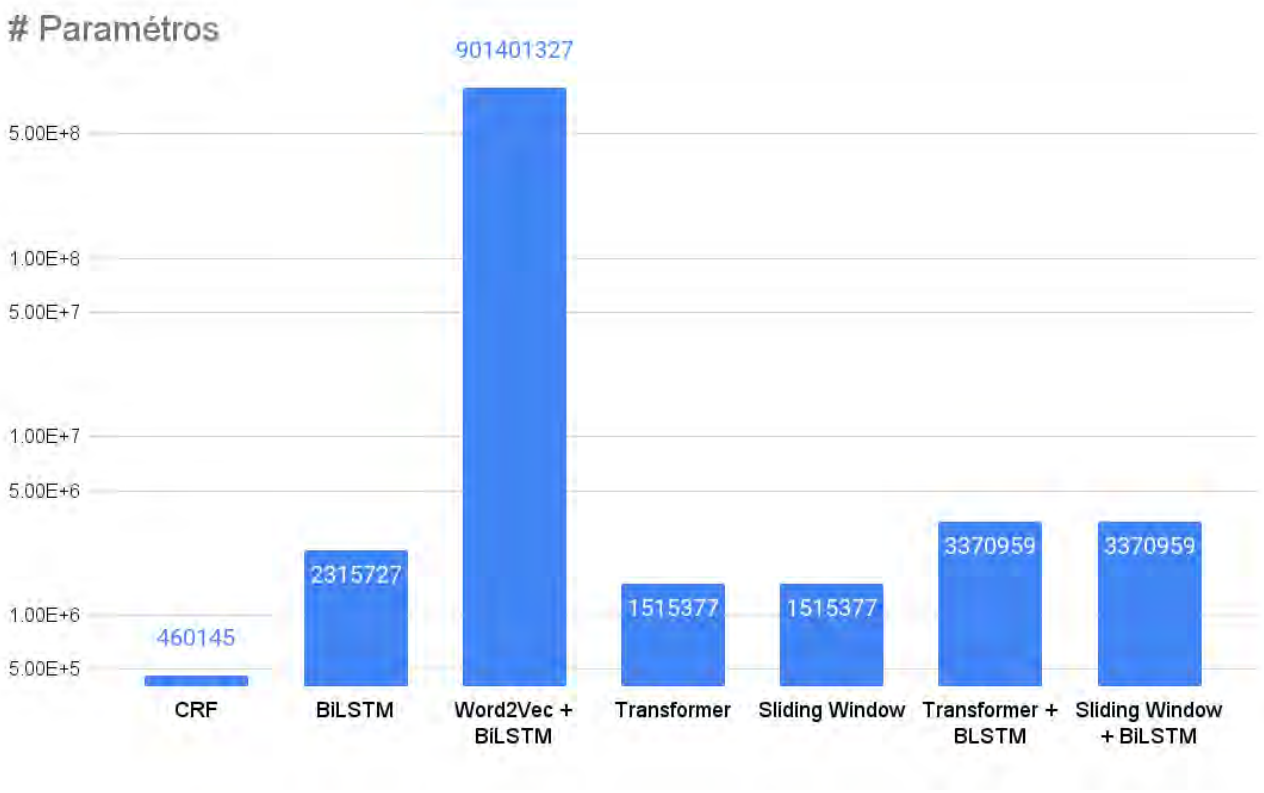


Figura 3.10: Comparación de la complejidad computacional de cada modelo con base en el número de parámetros

A continuación se presenta una descripción detallada y agrupada de la complejidad de cada modelo basado en el número de parámetros:

CRF: Modelo tiene el menor número de parámetros con un total de 460,145. Su menor complejidad computacional lo hace más eficiente en términos de tiempo de entrenamiento y recursos necesarios, lo que puede ser ventajoso en entornos con recursos limitados o cuando se necesita una rápida implementación.

BiLSTM: Con 2,315,727 parámetros, el modelo BiLSTM es significativamente más complejo que el CRF. Su capacidad para capturar contextos bidireccionales en las secuencias de datos justifica la complejidad adicional, resultando en un mayor rendimiento en tareas que requieren un entendimiento profundo del contexto.

Word2Vec + BiLSTM: Es el modelo más complejo en términos de parámetros, con un total de 901,401,327. La incorporación de Word2Vec como técnica de selección de características incrementa considerablemente el número de parámetros, lo que puede resultar en mayores requerimientos de tiempo y recursos computacionales durante el entrenamiento y la inferencia.

Transformer y Sliding Window: Ambos modelos cuentan con 1,515,377 parámetros. La arquitectura del Transformer basada en mecanismos de atención le permite capturar relaciones a largo plazo en los

datos, mejorando su rendimiento en tareas secuenciales complejas. Por otro lado, la técnica de Sliding Window es eficaz para capturar contextos locales.

Transformer + BiLSTM y Sliding Window + BiLSTM: Modelos que combinan las fortalezas de sus respectivas técnicas, resultando en 3,370,959 parámetros cada uno. La combinación de arquitecturas permite una captura más completa del contexto tanto a nivel local como global, a costa de una mayor complejidad computacional. Mejorando la robustez y precisión en la segmentación, aunque requieren mayores recursos computacionales.

3.7. El entrenamiento

A partir del corpus Giant, descrito en la sección 3.4.1, se generó el conjunto de entrenamiento y experimentación, utilizando un script en Python que transforma la referencia etiquetada en XML a formato CONLL-BIO.

Es importante recalcar que cada token que representa un signo de puntuación es marcado con la clase PUNC, dado que tales elementos son clave para distinguir los diferentes componentes de una cadena de referencia (Uddin, 2022). A continuación se muestra una referencia etiquetada con una clase como ejemplo:
Cita en cadena de texto crudo:

Ritchie, E. and Powell, Elmer Ellsworth (1907) Spinoza and Religion. The Philosophical Review, 16(3), p. 339. [online] Available from: http://dx.doi.org/10.2307/2177340

Cita en formato CONLL-BIO:

- Ritchie B-AUTHOR
- , B-PUNC
- E I-AUTHOR
- . B-PUNC
- and I-AUTHOR
- Powell I-AUTHOR
- , B-PUNC
- Elmer I-AUTHOR
- Ellsworth I-AUTHOR
- (B-PUNC
- 1907 B-YEAR

-) B-PUNC
- Spinoza B-TITLE
- and I-TITLE
- Religion I-TITLE
- . B-PUNC
- The B-CONTAINER-TITLE
- Philosophical I-CONTAINER-TITLE
- Review I-CONTAINER-TITLE
- , B-PUNC
- yo B-PUBLISHER
- mero I-PUBLISHER
- 16 B-VOLUME
- (B-PUNC
- 3 B-ISSUE
-) B-PUNC
- , B-PUNC
- p O
- . B-PUNC
- 339 B-PAGE
- . B-PUNC
- [B-PUNC
- online O
-] B-PUNC
- Available O
- from O
- : O
- http://dx.doi.org/10.2307/2177340 B-URL

Este conjunto de datos se compone de una muestra de 357391 registros, donde se representan todos los estilos de citación y todos los tipos de documentos[10], dividiéndolos proporcionalmente y de manera aleatoria, utilizando la librería SciKit-Learn[11] de Python y quedan de la siguiente manera:

1. 100069 (28 %) para el entrenamiento.
2. 12866 (3.6 %) para ajuste de hiperparámetros.
3. 12866 (3.6 %) para evaluar el rendimiento.
4. 231590 (64.8 %) para experimentación multiestilo (sección 4.2).

Para el entrenamiento se utilizaron los primeros tres conjuntos, mientras que el último se utiliza para experimentación.

Los hiperparámetros utilizados para los todos los modelos se muestran en la tabla 3.2:

Tabla 3.2: Parámetros de entrenamiento.

Parámetro	**Valor+CRF**
Tasa de aprendizaje	0.003
Tamaño de lote	1024
Máximo de épocas	150
Optimizador	AdamW
Paciencia	2

[10] El número de registros obedece a 10 ejemplos de cada combinación de estilo y tipo de documento, situación que no siempre se cumple, dando como resultado el número tan arbitrario.

[11] https://scikit-learn.org/

4 Experimentación y resultados

En el presente capítulo se detallan los experimentos realizados para evaluar y mejorar la segmentación de referencias bibliográficas.

Se presentan los resultados obtenidos mediante tres enfoques distintos: experimentos multiestilo, donde se exploran diferentes estilos de formato de referencias; experimento multilenguaje, que aborda la segmentación en tres idiomas; y experimento de tolerancia a errores e inconsistencias, centrado en evaluar la robustez de diferentes modelos frente a variaciones y errores comunes.

4.1. Evaluación del rendimiento

A continuación se presentan los resultados (ver figura 4.1) del entrenamiento y evaluación de cada modelo utilizando el conjunto de datos (12866 referencias) destinado a evaluar el rendimiento.

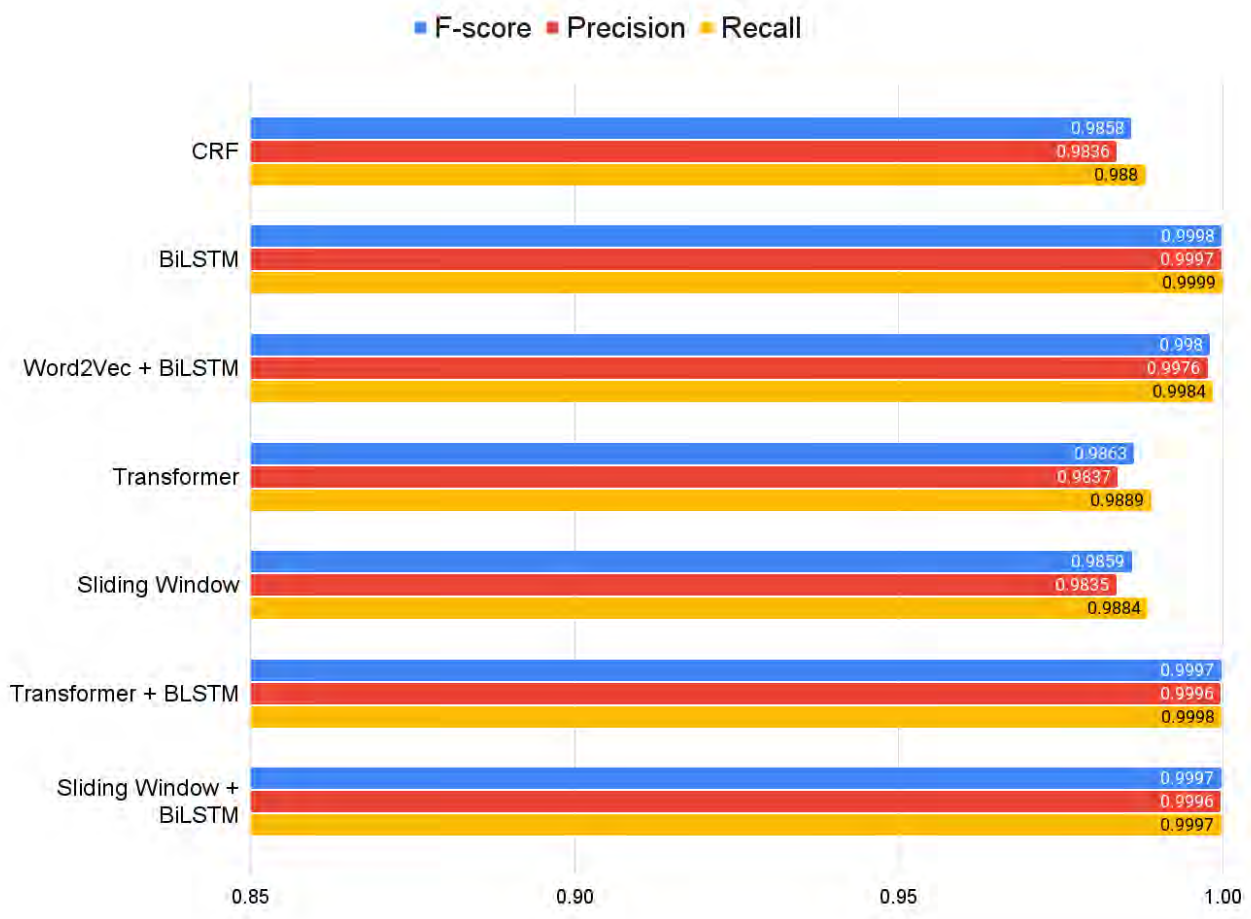


Figura 4.1: Comparación de resultados de rendimiento por modelo

Los resultados obtenidos muestran que los modelos BiLSTM y sus combinaciones con Word2Vec, Transformer y Sliding Window superan en rendimiento a los demás modelos evaluados. El BiLSTM puro alcanzó la mayor puntuación F-score de 0.9998, junto con de una precisión de 0.9997 y un recall de 0.9999, lo que indica una capacidad de segmentación casi perfecta de referencias bibliográficas en el conjunto de evaluación.

Las combinaciones de Transformer + BLSTM y Sliding Window + BiLSTM también presentaron resultados sobresalientes, con F-scores de 0.9997, precisiones de 0.9996 y 0.9996, y recalls de 0.9998 y 0.9997 respectivamente.

En contraste, los modelos basados únicamente en CRF, Transformer y Sliding Window mostraron un rendimiento menor, aunque todavía alto, con F-scores y valores de precisión y recall cercanos a 0.986, subrayando la ventaja comparativa de las arquitecturas basadas en BiLSTM para la tarea de segmentación de referencias.

Se debe aclarar que estos resultados corresponden a un subconjunto extraído del corpus GIANT el cual es la fuente del subconjunto de entrenamiento.

4.2. Experimentos multiestilo

Esta sección está dedicada a evaluar el rendimiento de los modelos, abarcando una diversidad de estilos de citación. Para lograrlo, se han diseñado dos experimentos que se describen a continuación.

4.2.1. Evaluación del conjunto de experimentación (GIANT)

A continuación, se presentan los resultados del primer experimento, para validar la capacidad multiestilo de los modelos (ver figura 4.2), se utilizó un conjunto más amplio (231590 referencias) que el de entrenamiento (100069 referencias), proveniente de Giant, con una muestra representativa aleatoria de los 1,564 estilos[1] contenidos en ese corpus.

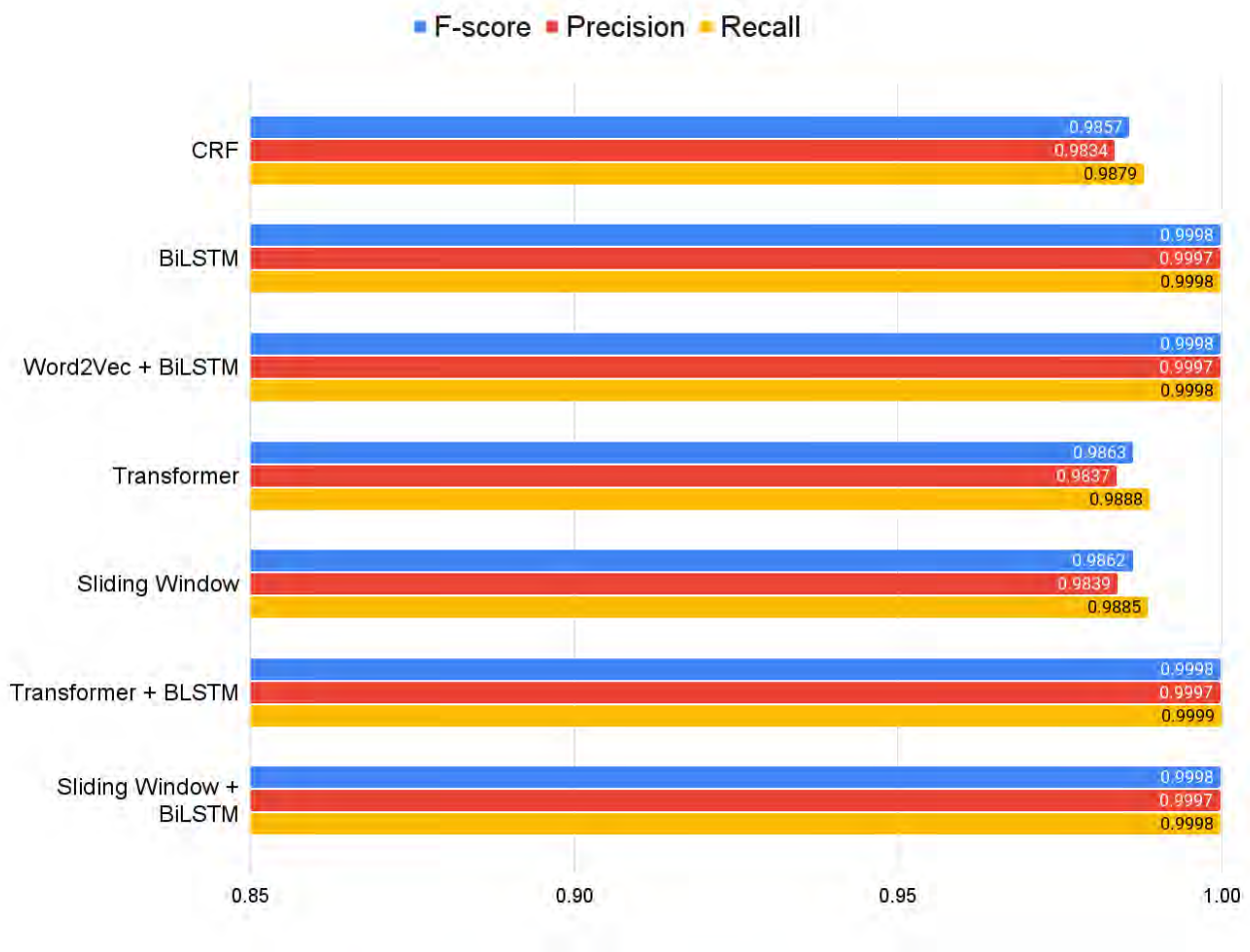


Figura 4.2: Comparación de resultados de rendimiento por modelo en el conjunto de datos de validación

[1] Correspondientes al catálogo de Citation Style Language https://github.com/citation-style-language/styles

Los resultados obtenidos muestran que, a pesar de la variación con respecto al tamaño del conjunto de datos de evaluación (sección 4.1), los modelos presentaron un desempeño casi idéntico en la segmentación de referencias bibliográficas en los diversos estilos que se incluyen en el corpus Giant.

4.2.2. Evaluación en siete estilos diferentes

El segundo experimento evalúa el rendimiento de los modelos en siete conjuntos de datos (ver figura 4.3), con 5000 registros y derivados del corpus Redalyc_eng, cada uno correspondiente a uno de los siguientes estilos de citación: APA6, VANCOUVER, HARVARD, CHICAGO, MLA, ISO y NLM.

A continuación se presentan los resultados del segundo experimento, destacando la capacidad de los modelos para segmentar con precisión referencias bibliográficas en una variedad de estilos de citación específicos.

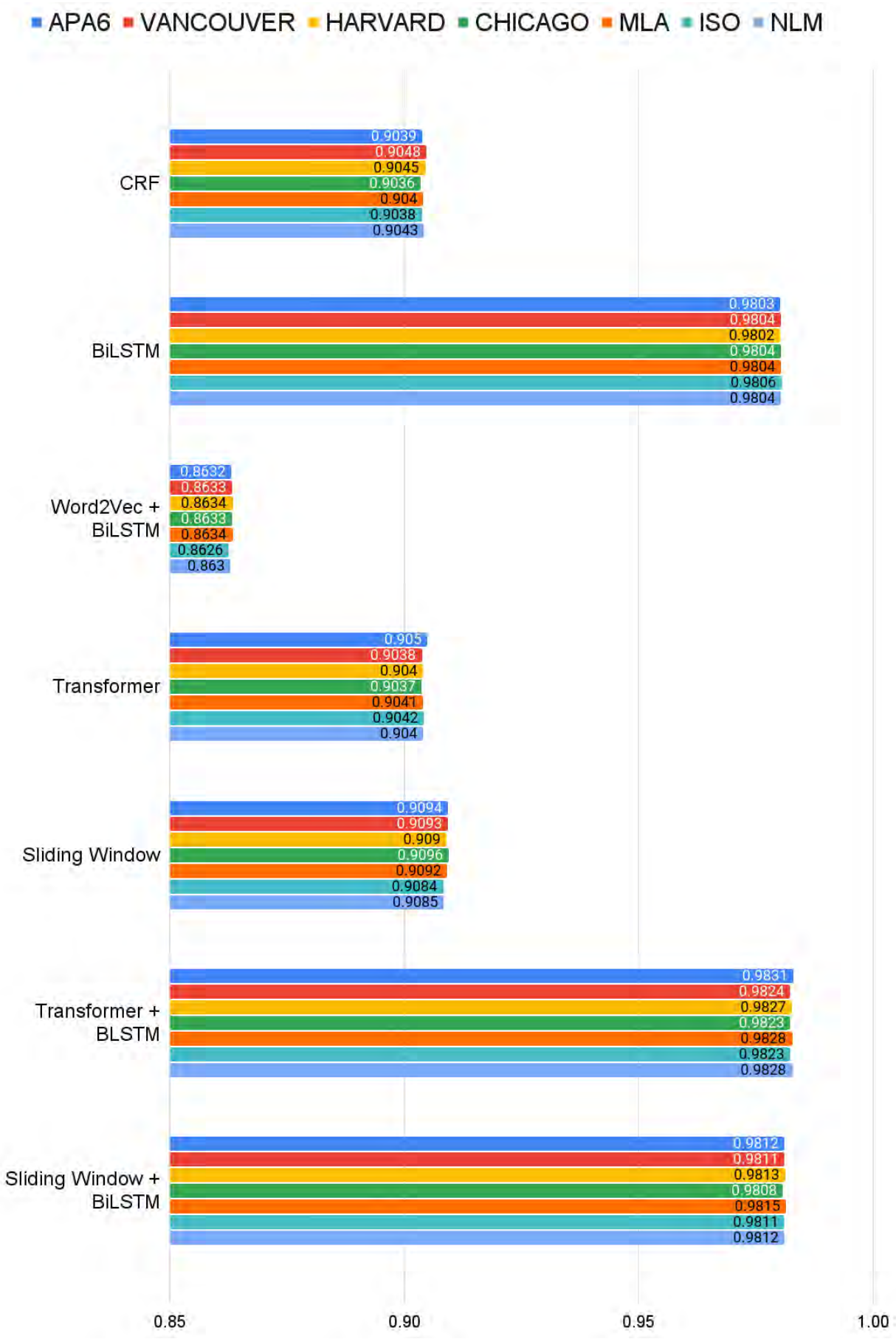


Figura 4.3: Comparación de resultados de rendimiento por modelo por estilo (F1-score)

En conjunto, los experimentos realizados en esta sección muestran que los modelos evaluados poseen una notable capacidad para segmentar con precisión referencias bibliográficas en una diversidad de estilos de citación.

Tanto el amplio conjunto de experimentación, como con los siete conjuntos de datos específicos por estilo, han revelado que los modelos, especialmente aquellos basados en BiLSTM y Transformer + BLSTM, mantienen un rendimiento consistente y alto, con F-scores que superan consistentemente el 98 %.

La consistencia en el rendimiento sugiere que los modelos son robustos y efectivos en la tarea de segmentación de referencias, adaptándose eficazmente a las variaciones estructurales y estilísticas presentes en distintas formas de citación bibliográfica. En contraste, el modelo Word2Vec + BiLSTM mostró un rendimiento inferior en todos estilos evaluados, además, resulta muy evidente su reducción en el rendimiento en los conjuntos de datos basados en Redalyc_eng.

Es relevante notar que la única diferencia significativa entre el modelo Word2Vec + BiLSTM y el modelo BiLSTM, radica en el enfoque utilizado para la selección de características, siendo Word2Vec específicamente utilizado en el primero y BPEmb en el segundo.

4.3. Experimento multilenguaje

Esta sección se centra en evaluar el rendimiento de los modelos en tres diferentes idiomas (español, inglés y francés), haciendo uso de los tres conjuntos de datos descritos en la sección 3.4.2 y que se distribuyen de la siguiente manera:

1. Redalyc_spa (Español) con 5000 registros.
2. Redalyc_eng (Inglés) con 5000 registros.
3. Redalyc_fra (Francés) con 2000 registros.

En la figura 4.4 se puede apreciar la evaluación del rendimiento de cada modelo en cada uno de los tres conjuntos de datos, cada uno corresponde a un idioma.

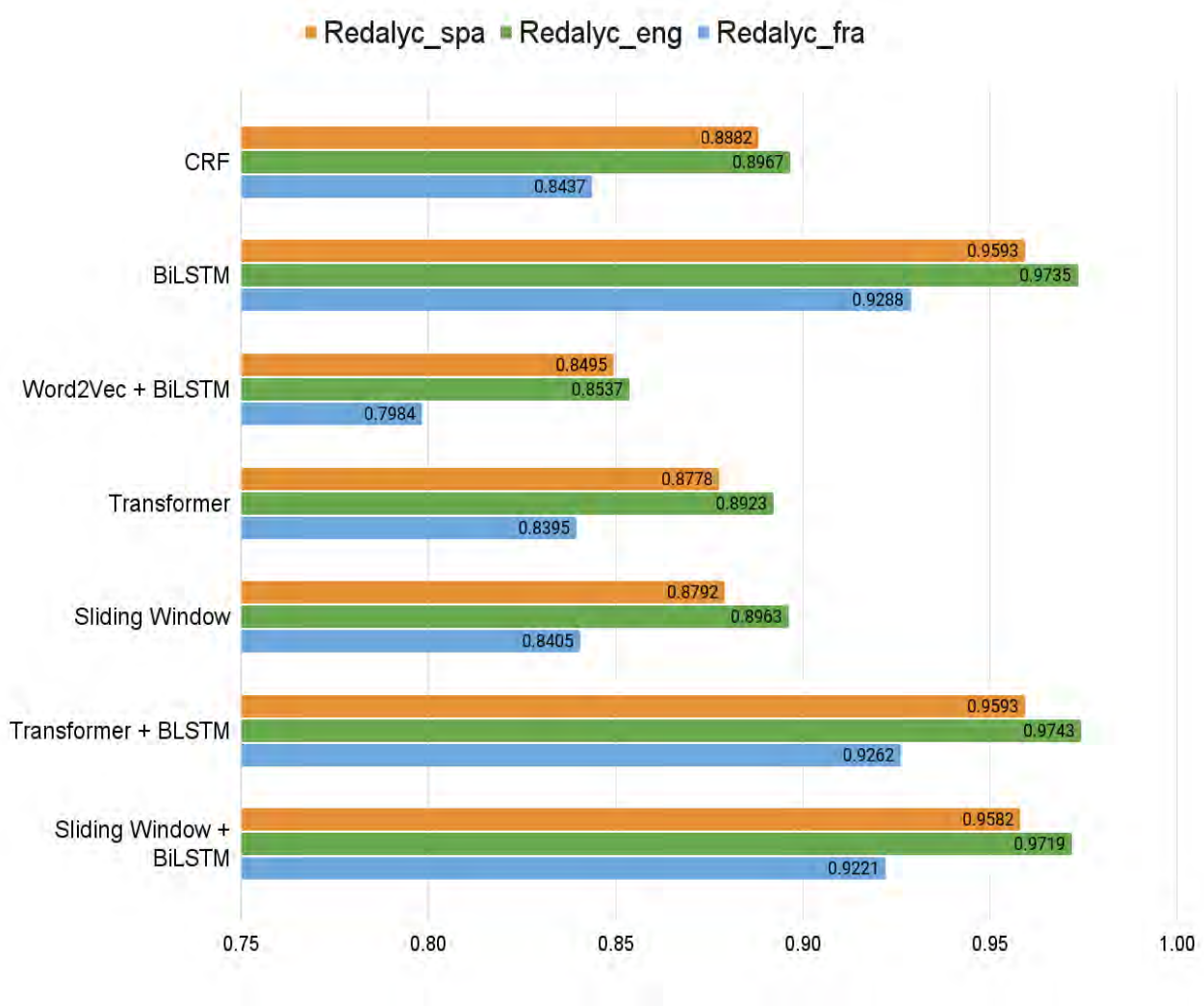


Figura 4.4: Comparación de resultados de rendimiento por modelo en tres idiomas diferentes (F1-Score)

Los resultados muestran que los modelos basados en BiLSTM y Transformer + BLSTM mantienen un rendimiento alto y consistente en los conjuntos de datos específicos por idioma, superando consistentemente el 97 % de F-score. Los modelos se adaptan efectivamente a las variaciones lingüísticas presentes en español, inglés y francés, como se evidencia en los conjuntos de datos Redalyc_spa, Redalyc_eng y Redalyc_fra.

En contraste, el modelo Word2Vec + BiLSTM mostró un rendimiento inferior en todos los idiomas evaluados, especialmente destacable en Redalyc_fra donde exhibe una notable reducción en el rendimiento. Esta diferencia sugiere que el enfoque utilizado para la selección de características, utilizando Word2Vec en lugar de BPEmb como en los otros modelos, afecta significativamente la capacidad de los modelos para segmentar referencias bibliográficas con precisión en entornos multilingües.

Es relevante destacar que la única diferencia sustancial entre el modelo Word2Vec + BiLSTM y los modelos BiLSTM y Transformer + BLSTM radica en el enfoque de selección de características. Subrayando la importancia de la elección del método adecuado de representación lingüística para mejorar el rendimiento y la adaptabilidad de los modelos en la segmentación de referencias bibliográficas.

Se debe resaltar el hecho que para el entrenamiento de los modelos, se utilizaron registros aleatorios

del corpus GIANT y en dicho corpus se estima que las referencias en Francés representan el 3.5 %, mientras que las referencias en español representan el 0.77 %, eso podría explicar el desempeño ligeramente superior en los resultados con el corpus Redalyc_eng.

4.4. Experimentos de tolerancia a omisiones e inconsistencias

Esta sección se enfoca en evaluar la capacidad de los modelos para manejar eficazmente omisiones e inconsistencias en la segmentación de referencias bibliográficas, utilizando el corpus CORA. Dicho corpus es ampliamente reconocido y utilizado en el estado del arte para la evaluación de sistemas de extracción de información bibliográfica (ver sección 3.4.1).

Una característica del corpus es que presenta diversas omisiones y variaciones estructurales por lo que lo hace adecuado para evaluar la tolerancia a estas, un caso representativo es el siguiente:

```
<author> M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M.
   Fahlman, O. Inganas and M.R. Andersson, </author> <container-
   title> J Appl. Phys., </container-title> <volume> 76, </volume><
   pages>893, </pages> <year> (1994). </year>
```

).

En la figura 4.5 se puede apreciar la evaluación del rendimiento de cada modelo en en el conjunto de datos CORA.

Los resultados obtenidos revelan diferencias significativas en el rendimiento de los modelos evaluados para manejar omisiones e inconsistencias en la segmentación de referencias bibliográficas, utilizando el corpus CORA. Entre los modelos analizados, se destaca que Sliding Window + BiLSTM y Transformer + BiLSTM alcanzan los F-scores más altos, con F-scores de 0.9618 y 0.9597 respectivamente, mostrando una mayor precisión en la segmentación de referencias.

En contraste, los modelos Sliding Window y Transformer muestran un rendimiento inferior cuando no se combinan con BiLSTM. Específicamente, Sliding Window obtiene un F-score de 0.8971, mientras que Transformer alcanza 0.8957, indicando una disminución significativa en la precisión de la segmentación cuando no se utiliza BiLSTM como componente.

Los hallazgos reportados subrayan la importancia de la arquitectura del modelo y la combinación adecuada de técnicas para lograr resultados precisos en la tarea de segmentación de referencias bibliográficas. La integración de BiLSTM desempeña un papel significativo en mejorar la capacidad de los modelos para manejar la complejidad estructural y las variaciones en los datos de entrada, mejorando así la precisión y la robustez del sistema.

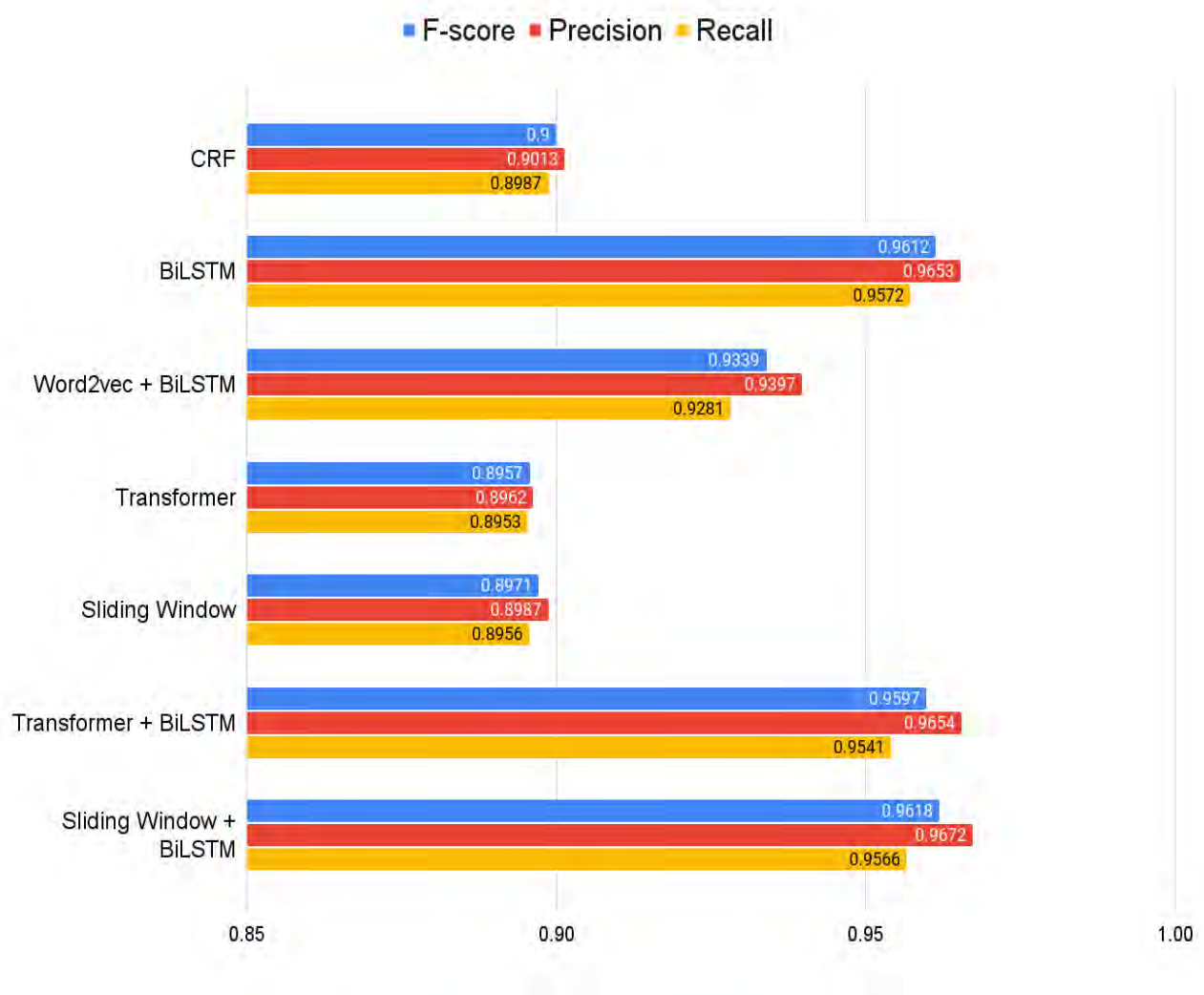


Figura 4.5: Comparación de resultados de rendimiento por modelo en el corpus CORA idiomas diferentes (F1-Score)

4.5. Análisis de resultados

A lo largo de la sección, se presentan y analizan los resultados obtenidos de los diversos experimentos realizados para evaluar la eficacia de los modelos en la tarea de segmentación de referencias bibliográficas.

4.5.1. Experimentos multiestilo

Los resultados obtenidos en los experimentos multiestilo destacan la robustez y consistencia de los modelos basados en BiLSTM y sus combinaciones con Transformer y Sliding Window. Los modelos BiLSTM, Transformer + BLSTM y Sliding Window + BLSTM lograron F-scores consistentemente altos, superando el 98 % en la segmentación de referencias bibliográficas a través de una amplia gama de estilos de citación.

Tal consistencia se mantiene incluso cuando se incrementa significativamente el tamaño del conjunto de validación, lo que sugiere una fuerte capacidad de generalización y adaptación a las variaciones estructurales inherentes a los diferentes estilos de citación.

Además de la validación con un conjunto amplio, se realizó un análisis más detallado evaluando los modelos en siete estilos específicos de citación: APA6, VANCOUVER, HARVARD, CHICAGO, MLA, ISO y NLM. Los resultados obtenidos en los experimentos muestran que los modelos basados en BiLSTM y sus combinaciones mantuvieron un rendimiento sobresaliente en todos los estilos, con F-scores que superaron el 98 %.

Mostrando su capacidad para segmentar con precisión referencias bibliográficas en una variedad de estilos subraya la versatilidad de las arquitecturas implementadas.

El análisis específico de los estilos revela que, aunque los modelos BiLSTM y sus combinaciones con Transformer y Sliding Window presentan un rendimiento alto y consistente en general, existen pequeñas variaciones en el rendimiento dependiendo del estilo de citación. Se debe notar que el modelo Transformer + BLSTM mostró una ligera ventaja en todos los estilos con respecto del modelo Sliding Window + BiLSTM.

En contraste, el modelo Word2Vec + BiLSTM mostró un rendimiento inferior en todos los estilos evaluados, inferior incluso al modelo CRF el cual carece de la fase de captura de contexto. La disminución en el rendimiento puede atribuirse a la diferencia en la fase de selección de características, donde el uso de Word2Vec en lugar de BPEmb parece afectar negativamente la precisión de la segmentación.

En particular, el modelo Transformer + BiLSTM mostró la mejor capacidad de adaptación con un F-score de 0.9831 en el estilo APA6 y 0.9828 en el estilo NLM. Los resultados superan significativamente al modelo basado en Word2Vec + BiLSTM, que es tomado como el representante del estado del arte, ya que no logró superar el 87 % en ninguno de los estilos evaluados.

Dicha diferencia en el rendimiento se considera como la medida de mejora en la segmentación multiestilo, evidenciando que las técnicas híbridas propuestas no solo incrementan la precisión general, sino que también aseguran una mayor consistencia en la segmentación, independientemente del estilo bibliográfico utilizado.

En general, los experimentos multiestilo revelan que los modelos basados en BiLSTM son particularmente efectivos para manejar la diversidad de estilos en la citación bibliográfica. La capacidad de mantener un rendimiento alto y consistente a través de diferentes estilos indica que los modelos están bien adaptados para aplicaciones en entornos con una amplia variedad de formatos de referencia.

Los resultados también sugieren que, aunque las combinaciones de modelos pueden ofrecer ventajas marginales en estilos específicos, la elección de la arquitectura de selección de características es fundamental para el rendimiento general del modelo.

Es relevante destacar que el modelo Word2Vec + BiLSTM presenta una considerable diferencia de rendimiento cuando se evalúa en un conjunto de datos proveniente de GIANT (el conjunto de entrenamiento proviene de Giant) y los conjuntos basados en Redalyc_eng, cosa que difiere en el modelo BiLSTM que es idéntico a excepción de la fase de selección de características. La diferencia muestra el impacto que tiene BPEmb en la capacidad de generalizar en un corpus distinto.

4.5.2. Experimento multilenguaje

Los resultados del experimento multilenguaje muestra que los modelos basados en BiLSTM y Transformer + BLSTM mantienen un rendimiento sobresaliente y consistente en español, inglés y

francés.

Los modelos lograron F-scores por encima del 97 %, evidenciando su capacidad para adaptarse eficazmente a las variaciones lingüísticas presentes en los datos. El rendimiento robusto en diferentes idiomas subraya la versatilidad y eficacia de las arquitecturas implementadas en la tarea de segmentación de referencias bibliográficas.

El modelo Word2Vec + BiLSTM, por otro lado, mostró un rendimiento inferior en todos los idiomas evaluados, especialmente en francés. Lo que sugiere que la técnica de selección de características basada en Word2Vec no es tan efectiva como BPEmb para tarea multilingües. La notable diferencia en el rendimiento destaca la importancia de la representación lingüística adecuada para mejorar la precisión y adaptabilidad de los modelos.

En términos de rendimiento multilenguaje, los modelos propuestos también superaron significativamente al estado del arte representado por Word2Vec + BiLSTM.

En el corpus Redalyc_spa, el modelo Transformer + BiLSTM alcanzó un F-score de 0.9593, mientras que Word2Vec + BiLSTM apenas logró un 0.8495. Esta tendencia se mantiene en inglés y francés, donde Transformer + BiLSTM alcanzó 0.9743 y 0.9262, respectivamente, destacándose como una solución eficaz para contextos multilingües.

Estos resultados subrayan la ventaja de utilizar BPEmb como técnica de selección de características, permitiendo una mejor adaptación lingüística en comparación con Word2Vec. En general, los experimentos muestran que los modelos híbridos propuestos no solo mejoran la segmentación en términos estilísticos, sino que también ofrecen una solución robusta y versátil en escenarios multilingües, aportando una clara ventaja sobre el estado del arte en esta tarea.

Es relevante volver a señalar que el corpus de entrenamiento (Giant) contiene una mayor proporción de referencias en inglés, lo cual podría explicar el desempeño ligeramente mayor en dicho idioma. No obstante, los modelos basados en BiLSTM y Transformer + BLSTM mostraron una capacidad sólida para manejar tanto español como francés.

4.5.3. Experimento de tolerancia a omisiones e inconsistencias

Los resultados del experimento de tolerancia a omisiones e inconsistencias, utilizando el corpus CORA, revelan que los modelos Sliding Window + BiLSTM y Transformer + BLSTM son los más efectivos para manejar las variaciones y omisiones en la segmentación de referencias bibliográficas. Estos modelos lograron los F-scores más altos, mostrando una mayor precisión y robustez frente a los desafíos presentados por las inconsistencias en los datos.

En contraste, los modelos basados únicamente en Sliding Window y Transformer mostraron un rendimiento significativamente inferior cuando no se combinan con BiLSTM, incluso el modelo Word2Vec + BiLSTM tuvo mejores resultados.

Mostrando que la inclusión de BiLSTM como componente principal es determinante para mejorar la capacidad de los modelos para manejar la complejidad estructural y las variaciones presentes en los datos de entrada.

La combinación adecuada de técnicas, como la integración de BiLSTM, parece ser un factor determinante para alcanzar un rendimiento óptimo en la segmentación de referencias bibliográficas.

En resumen, los resultados obtenidos subrayan la importancia de la arquitectura del modelo y la combinación de técnicas para mejorar la precisión y la robustez del sistema. Los modelos Sliding

Window + BiLSTM y Transformer + BLSTM se destacan como las opciones más efectivas para enfrentar las omisiones e inconsistencias en la segmentación de referencias, mostrando su capacidad para mantener un alto rendimiento en condiciones desafiantes.

Adicionalmente a los resultados de todas las categorías reportadas, es importante resaltar que el modelo CRF se mantuvo a la par e incluso superó ligeramente al modelo Sliding Window en todos los casos donde no se utilizó un conjunto de datos proveniente del corpus Giant. Esto sugiere que, a pesar de no tener la misma capacidad de captura de contexto que los modelos basados en BiLSTM, el modelo CRF sigue siendo una opción competitiva para la segmentación de referencias bibliográficas.

4.5.4. Comparación con el estado del arte

En esta sección se presenta una comparación entre los resultados obtenidos en este trabajo y los reportados en investigaciones previas sobre segmentación de referencias bibliográficas. Los estudios seleccionados incluyen los trabajos de Rodrigues Alves y cols. (2018), Prasad y cols. (2018) y Uddin (2022), los cuales representan avances relevantes en el uso de aprendizaje profundo para esta tarea.

Tabla 4.1: Comparación de F-scores entre los modelos evaluados y trabajos del estado del arte.

Modelo	**F-score**
CRF	0.900
BiLSTM	0.9612
Word2Vec + BiLSTM	0.9339
Transformer	0.8957
Sliding Window	0.8971
Transformer + BiLSTM	0.9597
Sliding Window + BiLSTM	**0.9618**
Rodrigues et al. (2018)	0.8966
Prasad et al. (2018)	0.9137
Uddin (2022)	0.942

Como muestra la Tabla 4.1, los modelos propuestos en este trabajo superan a los reportados en estudios anteriores. El enfoque basado en BiLSTM y sus combinaciones (Transformer + BiLSTM y Sliding Window + BiLSTM) alcanzan los F-scores más altos, superando consistentemente 0.959.

El trabajo de Rodrigues Alves y cols. (2018) utilizó una arquitectura BiLSTM + CRF con embeddings de Word2Vec, logrando un F-score de 0.8966 en segmentación. Aunque este modelo fue un avance relevante en su momento, los resultados de este trabajo muestran que la combinación de BiLSTM con Sliding Window mejora significativamente el rendimiento, alcanzando un F-score de 0.9618.

De manera similar, Prasad y cols. (2018) desarrollaron el modelo Neural Parscit, que combina LSTM y CRF, alcanzando un F-score de 0.9137. En comparación, el modelo Sliding Window + BiLSTM lo supera en rendimiento, subrayando la efectividad de las arquitecturas híbridas para capturar mejor el contexto.

Por último, Uddin (2022) aplicaron modelos basados en transformadores sobre un corpus de 220,000 referencias, logrando un F-score de 0.942. Si bien los transformadores demostraron ser efectivos, nuestros resultados sugieren que la combinación de BiLSTM con otras técnicas, como Sliding

Window (modelo BiLSTM + Sliding Window) , permite una segmentación más precisa, alcanzando un F-score de 0.9618.

Cabe destacar que los resultados alcanzados por Uddin (2022) usando Word2Vec superan al modelos Transformer y Sliding Window que utilizan BPEmb, esto sugiere que los transformadores funcionan mejor con una selección de características más complejas y que BPEmb.

En resumen, los modelos desarrollados en este trabajo destacan por su capacidad para superar las métricas del estado del arte. La integración de técnicas híbridas ha demostrado ser un enfoque efectivo para mejorar la segmentación de referencias, incluso frente a enfoques recientes basados en transformadores.

5 Conclusión

En esta tesis, se evaluaron diversos modelos para la segmentación de referencias bibliográficas, abarcando múltiples estilos de citación, idiomas y situaciones de tolerancia a omisiones e inconsistencias. Los resultados obtenidos proporcionan una comprensión integral de las fortalezas y limitaciones de cada modelo evaluado.

Los experimentos realizados mostraron que los modelos basados en BiLSTM, y sus combinaciones con Transformer y Sliding Window, destacan por su rendimiento consistente y robusto. Estos modelos superaron consistentemente el 98 % de F-score en tareas de segmentación de referencias bibliográficas a través de diversos estilos de citación y superaron el 92 % en entornos multilingües.

El rendimiento alcanzado resalta la capacidad de los modelos BiLSTM para capturar contextos bidireccionales y adaptarse eficazmente a variaciones estructurales en diferentes idiomas y formatos de referencia.

La diferencia significativa en el rendimiento entre los modelos Word2Vec + BiLSTM y BiLSTM subraya la importancia crítica de la selección de características. El enfoque basado en BPEmb mostró ser más efectivo y adaptable, mejorando la precisión de la segmentación en diversos estilos y contextos lingüísticos. Sugiriendo que la elección de la técnica de representación de características es un factor determinante en el rendimiento global del modelo.

Tomando en cuenta la complejidad computacional, resalta el hecho de que el modelo basado Word2Vec + BiLSTM, el cual es desproporcionalmente más grande que los siguientes en tamaño (267.4 veces) haya resultado el de menor rendimiento en todas las pruebas con conjuntos de datos provenientes de corpus diferentes al de entrenamiento.

Por otro lado, las combinaciones de Transformer + BiLSTM y Sliding Window + BiLSTM, las cuales ofrecen mejoras en rendimiento, también requieren recursos computacionales significativamente mayores.

En entornos con recursos limitados, los modelos más simples como CRF y BiLSTM, que mantienen una alta eficiencia sin sacrificar significativamente la precisión, pueden ser opciones más viables, se debe resaltar el hecho de que BiLSTM mantuvo una diferencia mínima de rendimiento con los modelos Transformer + BiLSTM y Sliding Window + BiLSTM, ratificándose como el modelo con la mejor relación costo/rendimiento.

La capacidad de los modelos para manejar omisiones e inconsistencias se evaluó utilizando el corpus CORA. Los resultados indicaron que las combinaciones de Sliding Window + BiLSTM y Transformer + BiLSTM son particularmente efectivas para esta tarea, alcanzando los F-scores más altos y demostrando una mayor precisión y robustez frente a variaciones en los datos.

Los hallazgos mencionados destacan la importancia de integrar técnicas que mejoren la tolerancia a errores estructurales, determinantes para aplicaciones en entornos reales donde la calidad de los datos puede variar.

El estudio también reveló que el modelo CRF, aunque más simple, mostró un rendimiento competitivo, especialmente en conjuntos de datos no provenientes del corpus Giant. Lo que sugiere que, para ciertas aplicaciones, modelos menos complejos pueden ofrecer un equilibrio adecuado entre eficiencia y costo computacional.

En la sección de 4.5.4, se destacó que los modelos propuestos en este trabajo superan a los reportados en estudios previos como los de Rodrigues Alves y cols. (2018), Prasad y cols. (2018) y Uddin (2022). La integración de BiLSTM con técnicas adicionales permitió alcanzar F-scores superiores, destacando especialmente el modelo Sliding Window + BiLSTM con un F-score de 0.9618.

Reflejando la importancia de combinar arquitecturas que capturen contexto sintáctico de forma eficiente. Los hallazgos subrayan que los enfoques híbridos presentan una ventaja clara frente a los modelos basados exclusivamente en transformadores para tareas que requieren segmentación precisa.

Es importante destacar que los modelos Transformer y Sliding Window, que se basan únicamente en transformadores, resultaron tener un rendimiento significativamente más bajo en comparación con los modelos que incluyen BiLSTM. Lo que puede explicarse por el hecho de que la tarea de segmentación es más sintáctica que semántica, y BiLSTM parece ser el componente clave en la captura de contexto suficiente para esta tarea.

En conjunto, los resultados del estudio proporcionan una guía para la selección y combinación de técnicas en la segmentación de referencias bibliográficas. Los modelos basados en BiLSTM, combinados con enfoques de representación de características adecuadas como BPEmb, se destacan como las opciones más robustas y precisas.

Sin embargo, la elección del modelo debe considerar el equilibrio entre complejidad computacional y rendimiento, adaptándose a las necesidades y limitaciones específicas del entorno de aplicación.

Los hallazgos antes mencionados contribuyen al avance del estado del arte en la segmentación de referencias bibliográficas, ofreciendo una base sólida para futuras investigaciones y desarrollos en estas áreas.

Con base todo lo expresado anteriormente, y en opinión del autor, se considera el modelo BiLSTM como el modelo que logra la mejora en la generalización de la subtarea de Segmentación e referencias, especialmente por su relación costo/rendimiento, lo que lo convierte en una opción factible para integrarse como un servicio en sistemas de cómputo de bibliotecas académicas.

5.1. Objetivos logrados

En la tabla 5.1 se muestra una breve descripción de cómo se cumplió con el objetivo general y con cada uno de los específicos.

Tabla 5.1: Logros por objetivo

Objetivo general	**Solución del objetivo**
Construir y evaluar un modelo que realice la subtarea de segmentación en la Minería de referencias, en diferentes idiomas y estilos de referencia.	Se construyeron diferentes modelos capaces de segmentar referencias bibliográficas, todos mostraron un rendimiento consistente y robusto. El modelo (Sliding Window + BiLSTM) con el puntaje mas alto logró superar el 98 % de F-score en múltiples estilos de citación y el 97 % en entornos multilingües.
Objetivos específicos	**Solución del objetivo**
Analizar el estado del arte en Minería de referencias e identificar los enfoques predominantes para su implementación.	Se revisaron 112 artículos de relevancia para el presente estudio, de los cuales solamente veintinueve cumplieron con los criterios de inclusión, es decir, el artículo debe abordar una o más de las subtareas de Minería de referencias.
Crear un corpus de referencias multilingüe y de diversos estilos bibliográficos.	Se desarrolló el corpus Redalyc a partir de referencias obtenidas de Redalyc.org, el único sitio encontrado con referencias debidamente etiquetadas en idioma y siete estilos de citación.
Determinar la arquitectura más tolerante a errores para la segmentación de referencias.	Se determinó que las arquitecturas de modelos basadas en combinaciones de Sliding Window + BiLSTM y Transformer + BiLSTM son las más tolerantes a errores para la segmentación de referencias. Estas combinaciones mostraron un mayor rendimiento y robustez frente a omisiones e inconsistencias en los datos, alcanzando los F-scores más altos en las evaluaciones realizadas, destacándose por su capacidad para manejar variaciones estructurales en la segmentación de referencias bibliográficas.
Establecer una comparación con el estado del arte que mantenga condiciones uniformes, mediante la recreación de la arquitectura mas representativa.	Se construyo el modelo Word2Vec + BiLSTM que no solamente emula a la arquitectura mas popular del estado del arte, si no que mantiene condiciones uniformes para una mejor comparación con el resto de los modelos construidos.

5.2. Aportaciones

- **Una revisión sobre la literatura**: El estudio realizó una amplia revisión de la literatura existente en el campo de la Minería de referencias bibliográficas, identificando y evaluando diversos enfoques y técnicas utilizadas anteriormente. La revisión literaria permite contextualizar los resultados obtenidos y compararlos con estudios previos, estableciendo así un avance significativo en el estado del arte de la RM (ver sección 6.1.3).

- **Creación del corpus Redalyc**: La creación del corpus Redalyc representa una contribución fundamental de este estudio, proporcionando un conjunto de datos robusto y diverso que

incluye referencias bibliográficas etiquetadas en múltiples idiomas y estilos de citación. El corpus facilita la evaluación de modelos en un contexto realista y variado para la comparación con otros estudios en el campo.

- **Creación de un modelo tolerante a errores, multilingüe y multiestilo**: El modelo no solamente integra técnicas de procesamiento de lenguaje natural y aprendizaje profundo, como BiLSTM y Transformer, sino que también presenta cualidades de tolerancia a los errores y variaciones estructurales inherentes a los datos bibliográficos. La capacidad de tolerancia a errores y adaptabilidad multilingüe amplía significativamente las aplicaciones prácticas del modelo en entornos académicos y científicos globales.

- **Evaluación de distintos enfoques**: El estudio evalúa y compara varias arquitecturas para la segmentación de referencias bibliográficas, incluyendo BiLSTM, Transformer, y combinaciones como Transformer + BLSTM.

  Esta evaluación exhaustiva permite identificar las fortalezas y debilidades de cada enfoque en términos de precisión, robustez y adaptabilidad a diferentes estilos y idiomas. Los resultados enriquecen el conocimiento técnico y metodológico en el campo.

- **Construcción de modelos con distinto costo computacional**: El estudio evalúa el rendimiento de modelos con diferentes niveles de costo computacional, desde modelos más simples como CRF hasta modelos más complejos como BiLSTM, Transformer y combinaciones de los mismos. La diversidad en los modelos permite ilustrar cómo el rendimiento de la segmentación de referencias bibliográficas puede equilibrarse con los recursos computacionales disponibles.

5.3. Trabajo futuro

Para continuar avanzando en el campo de la segmentación de referencias bibliográficas, se proponen tres líneas de investigación que podrían ampliar y mejorar los resultados obtenidos en este estudio:

1. **Explorar diversas modificaciones a la capa atencional diferentes a la ventana deslizante**: Se sugiere investigar y desarrollar nuevas variantes de la capa atencional que puedan mejorar la precisión y eficiencia de la segmentación. Con la posibilidad de incluir métodos como la atención multinivel, atención dirigida o adaptativa, entre otros, que podrían adaptarse a las características específicas de los datos bibliográficos.

2. **Explorar la segmentación multilingüe con una mayor variedad de idiomas**: Ampliar el estudio a una gama más amplia de idiomas podría revelar cómo diferentes estructuras lingüísticas afectan el rendimiento de los modelos. Incluyendo la evaluación en idiomas con características morfológicas y sintácticas distintas, lo cual es necesario para validar la robustez y generalización de los modelos en entornos multilingües.

3. **Explorar otras arquitecturas más complejas como LLM o Modelos Amplios de Lenguaje Enmascarado (BERT)**: Investigar la aplicación de arquitecturas avanzadas como LLM (Large Language Models) o BERT para la tarea de segmentación de referencias bibliográficas podría proporcionar nuevas perspectivas y mejoras significativas en la precisión y capacidad de adaptación de nuevos modelos.

Estas arquitecturas son conocidas por su capacidad para capturar contextos complejos y representaciones lingüísticas más profundas, lo cual podría beneficiar la tarea de manejar las variaciones estructurales presentes en las referencias bibliográficas.

Las líneas de investigación antes mencionadas proponen avanzar hacia sistemas más precisos, adaptables y eficientes para la segmentación de referencias bibliográficas en diversos contextos.

Referencias

Agrawal, K., Mittal, A., y Pudi, V. (2019). Scalable, Semi-Supervised Extraction of Structured Information from Scientific Literature [proceedings-article]. En (pp. 11–20). Association for Computational Linguistics. Descargado de http://dx.doi.org/10.18653/v1/w19-2602 doi: 10.18653/v1/w19-2602

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., y Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP [proceedings-article]. En *NAACL 2019, 2019 annual conference of the north american chapter of the association for computational linguistics (demonstrations)* (pp. 54–59). Association for Computational Linguistics. Descargado de http://dx.doi.org/10.18653/v1/w18-2505 doi: 10.18653/v1/w18-2505

Almeida, F., y Xexéo, G. (2019). Word embeddings: A survey [journal-article]. *arXiv preprint arXiv:1901.09069, 89*. Descargado de http://dx.doi.org/10.1016/j.inffus.2022.08.024 doi: 10.1016/j.inffus.2022.08.024

Anzaroot, S., y McCallum, A. (2013). A new dataset for fine-grained citation field extraction [component]. *ICML 2013 Workshop on Peer Reviewing and Publishing Models*. Descargado de http://dx.doi.org/10.7717/peerj-cs.2011/supp-3 doi: 10.7717/peerj-cs.2011/supp-3

Bender, E. (2019). The #benderrule: On naming the languages we study and why it matters [journal-article]. *The Gradient*. https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/. Descargado de http://dx.doi.org/10.36078/987654329 doi: 10.36078/987654329

Bhardwaj, A., Mercier, D., Dengel, A., y Ahmed, S. (2017). Deepbibx: Deep learning for image based bibliographic data extraction [book-chapter]. En *International conference on neural information processing* (pp. 286–293). Springer International Publishing. Descargado de http://dx.doi.org/10.1007/978-3-319-70096-0_30 doi: 10.1007/978-3-319-70096-0_30

Boukhers, Z., Ambhore, S., y Staab, S. (2019, jun). An end-to-end approach for extracting and segmenting high-variance references from PDF documents [proceedings-article]. En *Proceedings of the acm/ieee joint conference on digital libraries* (Vol. 2019-June, pp. 186–195). IEEE. Descargado de http://dx.doi.org/10.1109/jcdl.2019.00035 doi: 10.1109/jcdl.2019.00035

Chen, C.-C., Yang, K.-H., Kao, H.-Y., y Ho, J.-M. (2008). Bibpro: A citation parser based on sequence alignment techniques [proceedings-article]. En *22nd international conference on advanced information networking and applications-workshops (aina workshops 2008)* (pp. 1175–1180). IEEE. Descargado de http://dx.doi.org/10.1109/waina.2008.125 doi: 10.1109/waina.2008.125

Choi, W., Yoon, H. M., Hyun, M. H., Lee, H. J., Seol, J. W., Lee, K. D., ... Kong, H. (2023, 1). Building an annotated corpus for automatic metadata extraction from multilingual journal article references [journal-article]. *PLOS ONE*, *18*, e0280637. Descargado de http://dx.doi.org/10.1371/journal.pone.0280637 doi: 10.1371/journal.pone.0280637

Chollet, F. (2017). Deep learning with python, vol. 1. *Greenwich, CT: Manning Publications CO*. Descargado de http://dx.doi.org/10.4135/9781526493446 doi: 10.4135/9781526493446

Chowdhury, G. G. (1999). Template mining for information extraction from digital documents [proceedings-article]. *Library Trends*, *48*(1), 182–208. Descargado de http://dx.doi.org/10.62919/jhgh2876 doi: 10.62919/jhgh2876

Cortez, E., da Silva, A. S., Gonçalves, M. A., Mesquita, F., y de Moura, E. S. (2007). Flux-cim: flexible unsupervised extraction of citation metadata [proceedings-article]. En *Proceedings of the 7th acm/ieee-cs joint conference on digital libraries* (pp. 215–224). ACM. Descargado de http://dx.doi.org/10.1145/1255175.1255219 doi: 10.1145/1255175.1255219

Councill, I. G., Giles, C. L., y Kan, M.-Y. (2008). ParsCit: An open-source CRF Reference String and Logical Document Structure Parsing Package [journal-article]. En *Proceedings of the 6th international conference on language resources and evaluation* (Vol. 19, pp. 661–667). Springer Science and Business Media LLC. Descargado de http://dx.doi.org/10.1007/s00799-018-0242-1 doi: 10.1007/s00799-018-0242-1

Day, M.-Y., Tsai, R. T.-H., Sung, C.-L., Hsieh, C.-C., Lee, C.-W., Wu, S.-H., ... Hsu, W.-L. (2007). Reference metadata extraction using a hierarchical knowledge representation framework [journal-article]. *Decision Support Systems*, *43*(1), 152–167. Descargado de http://dx.doi.org/10.1016/j.dss.2006.08.006 doi: 10.1016/j.dss.2006.08.006

Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding [component]. *arXiv preprint arXiv:1810.04805*. Descargado de http://dx.doi.org/10.1109/tim.2024.3374300/mm1 doi: 10.1109/tim.2024.3374300/mm1

Ding, Y., Chowdhury, G., y Foo, S. (1999). Template Mining for the Extraction of Citation From Digital Documents [proceedings-article]. *Proceedings of the Second Asian Digital Library Conference Taiwan*, *639798*(February 2013), 47–62. Descargado de http://dx.doi.org/10.62919/jhgh2876 doi: 10.62919/jhgh2876

Galicia, S., Gelbukh, A., y Bolshakov, I. (1998). Diccionario de patrones de manejo sintáctico para análisis de textos en español. [dissertation]. *Procesamiento del Lenguaje Natural*, *23*. Descargado de http://dx.doi.org/10.31390/gradschool_theses.5883 doi: 10.31390/gradschool_theses.5883

Graves, A., y Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm networks [proceedings-article]. En *Proceedings. 2005 ieee international joint conference on neural networks, 2005.* (Vol. 4, pp. 2047–2052). IEEE. Descargado de http://dx.doi.org/10.1109/ijcnn.2005.1556215 doi: 10.1109/ijcnn.2005.1556215

Grennan, M., y Beel, J. (2020, apr). Synthetic vs. Real Reference Strings for Citation Parsing, and the Importance of Re-training and Out-Of-Sample Data for Meaningful Evaluations: Experiments with GROBID, GIANT and Cora [dataset]. *arxiv.org*, 1–7. Descargado de http://dx.doi.org/10.32614/cran.package.randomuseragent doi: 10.32614/cran.package.randomuseragent

Grennan, M., Schibel, M., Collins, A., y Beel, J. (2019). Giant: The 1-billion annotated synthetic bibliographic-reference-string dataset for deep citation parsing [book-chapter]. En *Ceur workshop proceedings* (Vol. 2563, pp. 260–271). Springer Berlin Heidelberg. Descargado de http://dx.doi.org/10.1007/978-3-642-03547-0_10 doi: 10.1007/978-3-642-03547-0_10

Gupta, T. K., y Raza, K. (2019). Optimization of ann architecture: a review on nature-inspired techniques [book-chapter]. *Machine learning in bio-signal analysis and diagnostic imaging*, 159–182. Descargado de http://dx.doi.org/10.1016/b978-0-12-816086-2.00007-2 doi: 10.1016/b978-0-12-816086-2.00007-2

Heinzerling, B., y Strube, M. (2018, May). BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages [dataset]. En *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA). Descargado de http://dx.doi.org/10.32614/cran.package.sentencepiece doi: 10.32614/cran.package.sentencepiece

Hetzner, E. (2008). A simple method for citation metadata extraction using hidden markov models [proceedings-article]. En *Proceedings of the 8th acm/ieee-cs joint conference on digital libraries* (pp. 280–284). ACM. Descargado de http://dx.doi.org/10.1145/1378889.1378937 doi: 10.1145/1378889.1378937

Hochreiter, S., y Schmidhuber, J. (1997). Long short-term memory [component]. *Neural computation*, *9*(8), 1735–1780. Descargado de http://dx.doi.org/10.7717/peerjcs.216/fig-4 doi: 10.7717/peerjcs.216/fig-4

Hsieh, Y.-L., Liu, S.-H., Yang, T.-H., Chen, Y.-H., Chang, Y.-C., Hsieh, G., … Hsu, W.-L. (2014). A frame-based approach for reference metadata extraction [book-chapter]. En *International conference on technologies and applications of artificial intelligence* (pp. 154–163). Springer International Publishing. Descargado de http://dx.doi.org/10.1007/978-3-319-13987-6_15 doi: 10.1007/978-3-319-13987-6_15

Jain, V., Baliyan, N., y Kumar, S. (2023). Machine learning approaches for entity extraction from citation strings [book-chapter]. En *International conference on information technology* (pp. 287–297). Springer Nature Singapore. Descargado de http://dx.doi.org/10.1007/978-981-99-5997-6_25 doi: 10.1007/978-981-99-5997-6_25

Kashyap, A. R., y Kan, M. Y. (2020, apr). *SciWING – A software toolkit for scientific document processing* [proceedings-article]. Association for Computational Linguistics. Descargado de http://dx.doi.org/10.18653/v1/2020.sdp-1.13 doi: 10.18653/v1/2020.sdp-1.13

Ketkar, N. (2017). *Deep learning with python* (1.ª ed.). New York, NY: APRESS. Descargado de http://dx.doi.org/10.4135/9781526493446 doi: 10.4135/9781526493446

Körner, M. (2017). Reference string extraction using line-based conditional random fields [book-chapter]. *arXiv*(n/a). Descargado de http://dx.doi.org/10.1007/978-3-319-67162-8_15 doi: 10.1007/978-3-319-67162-8_15

Körner, M., Ghavimi, B., Mayr, P., Hartmann, H., y Staab, S. (2017). Evaluating reference string extraction using line-based conditional random fields: A case study with german language publications [book-chapter]. En *European conference on advances in databases and information systems* (pp. 137–

145). Springer International Publishing. Descargado de http://dx.doi.org/10.1007/978-3-319-67162-8_15 doi: 10.1007/978-3-319-67162-8_15

Lafferty, J., McCallum, A., Pereira, F., y cols. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data [proceedings-article]. En *Icml* (Vol. 1, p. 3). ACM Press. Descargado de http://dx.doi.org/10.1145/1015330.1015422 doi: 10.1145/1015330.1015422

LeCun, Y., Bengio, Y., y Hinton, G. (2015, May). Deep learning [other]. *Nature*, *521*(7553), 436–444. Descargado de http://dx.doi.org/10.1017/9781108955652.016 doi: 10.1017/9781108955652.016

Ling, W., Luís, T., Marujo, L., Astudillo, R. F., Amir, S., Dyer, C., ... Trancoso, I. (2015). Finding function in form: Compositional character models for open vocabulary word representation [proceedings-article]. *arXiv preprint arXiv:1508.02096*. Descargado de http://dx.doi.org/10.18653/v1/d15-1176 doi: 10.18653/v1/d15-1176

Lopez, P. (2009). Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications [book-chapter]. En *International conference on theory and practice of digital libraries* (pp. 473–474). Springer Berlin Heidelberg. Descargado de http://dx.doi.org/10.1007/978-3-642-04346-8_62 doi: 10.1007/978-3-642-04346-8_62

López Piñero, J. M. (1972). El análisis estadístico y sociométrico de la literatura científica [journal-article]. *Valencia: Centro de documentación e informática médica*, *197*, 2007–2012. Descargado de http://dx.doi.org/10.1344/bid2018.41.10 doi: 10.1344/bid2018.41.10

Mikolov, T., Chen, K., Corrado, G., Dean, J., Sutskever, L., y Zweig, G. (2013). word2vec [component]. *URL https://code. google. com/p/word2vec*, *22*, 795. Descargado de http://dx.doi.org/10.7717/peerj-cs.1248/supp-3 doi: 10.7717/peerj-cs.1248/supp-3

Morán Reyes, A. A. (2020). Las bibliotecas especializadas frente a la pandemia de la covid-19 (special libraries and the covid-19 pandemic) [book-chapter]. *Filosofía y Letras: Revista de la Facultad de Filosofía y Letras*(4). Descargado de http://dx.doi.org/10.18800/9786123177348.001 doi: 10.18800/9786123177348.001

Ojokoh, B., Zhang, M., y Tang, J. (2011). A trigram hidden markov model for metadata extraction from heterogeneous references [journal-article]. *Information Sciences*, *181*(9), 1538–1551. Descargado de http://dx.doi.org/10.1016/j.ins.2011.01.014 doi: 10.1016/j.ins.2011.01.014

Padrón Guillén, J. (1996). Análisis del discurso e investigación social: Temas para seminario [journal-article]. *301 1 CIC-UCAB/0255 20040126 GR, LXIV*. Descargado de http://dx.doi.org/10.3989/ris.2006.i44.25 doi: 10.3989/ris.2006.i44.25

Patro, S. (2012). *Data Clustering and Cleansing for Bibliography Analysis* (book-chapter, University of New South Wales). doi: 10.1201/9781420034912.bmatt1

Pena-Rocha, M., Gomez-Crisostomo, M. R., Guerrero-Bote, V. P., y de Moya-Anegón, F. (2024). Bibliometrics effects of a new item-by-item classification system based on reference reclassification [dissertation]. The University of Iowa. Descargado de http://dx.doi.org/10.17077/etd.59agk7vw doi: 10.17077/etd.59agk7vw

Peng, F., y McCallum, A. (2006a). Information extraction from research papers using conditional random fields [journal-article]. *Information Processing and Management*, *42*(4), 963–979. Descargado de http://dx.doi.org/10.1016/j.ipm.2005.09.002 doi: 10.1016/j.ipm.2005.09.002

Peng, F., y McCallum, A. (2006b). Information extraction from research papers using conditional random fields [journal-article]. *Information processing & management*, *42*(4), 963–979. Descargado de http://dx.doi.org/10.1016/j.ipm.2005.09.002 doi: 10.1016/j.ipm.2005.09.002

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., y Zettlemoyer, L. (2018, June). Deep contextualized word representations [proceedings-article]. En M. Walker, H. Ji, y A. Stent (Eds.), *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics. Descargado de http://dx.doi.org/10.1109/inista52262.2021.9548419 doi: 10.1109/inista52262.2021.9548419

Prasad, A., Kaur, M., y Kan, M. Y. (2018). Neural ParsCit: a deep learning-based reference string parser [journal-article]. *International Journal on Digital Libraries*, *19*(4), 323–337. Descargado de http://dx.doi.org/10.1007/s00799-018-0242-1 doi: 10.1007/s00799-018-0242-1

Rizvi, S. T. R., Dengel, A., y Ahmed, S. (2019, dec). DeepBiRD: An Automatic Bibliographic Reference Detection Approach [book-chapter]. *arxiv*(n/a). Descargado de http://dx.doi.org/10.4324/9781315178899-8 doi: 10.4324/9781315178899-8

Rodrigues, F., Pereira, F., y Ribeiro, B. (2014). Sequence labeling with multiple annotators. *Machine learning*, *95*(2), 165–181.

Rodrigues Alves, D., Colavizza, G., y Kaplan, F. (2018, jul). Deep Reference Mining From Scholarly Literature in the Arts and Humanities [journal-article]. *Frontiers in Research Metrics and Analytics*, *3*. Descargado de http://dx.doi.org/10.3389/frma.2018.00021 doi: 10.3389/frma.2018.00021

Romanello, M., Boschetti, F., y Crane, G. (2009). Citations in the digital library of classics: extracting canonical references by using conditional random fields [proceedings-article]. En *Proceedings of the 2009 workshop on text and citation analysis for scholarly digital libraries (nlpir4dl)* (pp. 80–87). Association for Computational Linguistics. Descargado de http://dx.doi.org/10.3115/1699750.1699763 doi: 10.3115/1699750.1699763

Russell, S. J., y Norvig, P. (2004). *Inteligencia artificial: un enfoque moderno* (Vol. 4) (journal-article n.os 04; Q335, R8y 2004.). Puerto Madero Editorial Académica. Descargado de http://dx.doi.org/10.55204/trc.v4i2.e395 doi: 10.55204/trc.v4i2.e395

Santos, E. A. d., Peroni, S., y Mucheroni, M. L. (2023). An analysis of citing and referencing habits across all scholarly disciplines: approaches and trends in bibliographic referencing and citing practices [journal-article]. *Journal of Documentation*, *79*(7), 196–224. Descargado de http://dx.doi.org/10.1108/jd-10-2022-0234 doi: 10.1108/jd-10-2022-0234

Tkaczyk, D., Collins, A., Sheridan, P., y Beel, J. (2018). Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers [proceedings-article]. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 99–108. Descargado de http://dx.doi.org/10.1145/3197026.3197048 doi: 10.1145/3197026.3197048

Tkaczyk, D., Gupta, R., Cinti, R., y Beel, J. (2018). ParsRec: A novel meta-learning approach to recommending bibliographic reference parsers [proceedings-article]. En *Ceur workshop proceedings* (Vol. 2259, pp. 162–173). ACM. Descargado de http://dx.doi.org/10.1145/3197026.3197048 doi: 10.1145/3197026.3197048

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., y Bolikowski, Ł. (2015, jul). CERMINE: Automatic extraction of structured metadata from scientific literature [journal-article]. *International Journal on Document Analysis and Recognition*, *18*(4), 317–335. Descargado de http://dx.doi.org/10.1007/s10032-015-0249-8 doi: 10.1007/s10032-015-0249-8

Uddin, M. S. (2022). *Transparscit: A transformer-based citation parser trained on large-scale synthesized data* (component, Old Dominion University). doi: 10.1109/jbhi.2023.3288768/mm1

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., ... Polosukhin, I. (2017). Attention is all you need [other]. *arxiv*. Descargado de http://dx.doi.org/10.5040/9781350101272.00000005 doi: 10.5040/9781350101272.00000005

Wikipedia. (2021a). *Aprendizaje automático — wikipedia, la enciclopedia libre* [journal-article]. Universidad Nacional de Cordoba. Descargado de http://dx.doi.org/10.31048/1852.4826.v17.n1.43539 ([Internet; descargado 30-diciembre-2021]) doi: 10.31048/1852.4826.v17.n1.43539

Wikipedia. (2021b). *Procesamiento de lenguajes naturales — wikipedia la enciclopedia libre* [journal-article]. Universidad Nacional de Cordoba. Descargado de http://dx.doi.org/10.31048/1852.4826.v17.n1.43539 ([Internet; descargado 29-diciembre-2021]) doi: 10.31048/1852.4826.v17.n1.43539

Yang, T. H., Hsieh, Y. L., Liu, S. H., Chang, Y. C., y Hsu, W. L. (2020, aug). A flexible template generation and matching method with applications for publication reference metadata extraction [journal-article]. *Journal of the Association for Information Science and Technology*, *72*, asi.24391. Descargado de http://dx.doi.org/10.1002/asi.24391 doi: 10.1002/asi.24391

Yin, P., Zhang, M., Deng, Z., y Yang, D. (2004). Metadata extraction from bibliographies using bigram hmm [book-chapter]. En *International conference on asian digital libraries* (pp. 310–319). Springer Berlin Heidelberg. Descargado de http://dx.doi.org/10.1007/978-3-540-30544-6_33 doi: 10.1007/978-3-540-30544-6_33

Zhang, X., Zou, J., Le, D. X., y Thoma, G. R. (2011). A structural SVM approach for reference parsing [journal-article]. *BMC Bioinformatics*, *12*(SUPPL. 3). Descargado de http://dx.doi.org/10.1186/1471-2105-12-s3-s7 doi: 10.1186/1471-2105-12-s3-s7

Zou, J., Le, D., y Thoma, G. R. (2010). Locating and parsing bibliographic references in html medical articles [journal-article]. *International Journal on Document Analysis and Recognition (IJDAR)*, *13*(2), 107–119. Descargado de http://dx.doi.org/10.1007/s10032-009-0105-9 doi: 10.1007/s10032-009-0105-9

6 Anexos

6.1. Actividades académicas adicionales

6.1.1. Cursos impartidos

- Taller Introducción al procesamiento del lenguaje natural - Facultad de Comunicación, Universidad de la Habana, La Habana Cuba, ver figura 6.1.

Figura 6.1: Constancia Taller NLP CUBA

- Taller Introducción a la Ciencia de Datos - Facultad de Ciencias Sociales, Escuela de Historia, Sección Archivística, Universidad de Costa Rica, San Jose Costa Rica, 20/03/2023, ver figura 6.2

UNIVERSIDAD DE COSTA RICA

Otorga el presente

CERTIFICADO DE AGRADECIMIENTO A

Rodrigo Cuéllar Hidalgo

Por sus aportes académicos en el Taller: "Ciencia de datos" en el Foro Internacional de Evaluación de Documentos en la Era Digital, FIED-2023

Impartido el 20 de marzo de 2023, en el marco del proyecto:
ED-2794 Formación Archivística Continua de la Carrera de Archivística de la Escuela de Historia

Firmado en la Ciudad Universitaria Rodrigo Facio, a los 20 días del mes de marzo de 2023.

M.Sc. Claudio Vargas Arias
Director, Escuela de Historia

ESCUELA DE HISTORIA
UNIVERSIDAD DE COSTA RICA

M.Ls. María Teresa Bermúdez Muñoz
Coordinadora, Carrera de Archivística

Figura 6.2: Constancia Taller Ciencia de Datos Costa Rica

6.1.2. Eventos académicos

- Presentación en Electronic Theses and Dissertations, ETD[1] 2021 Open Scholarship in a Post-Pandemic World 16/11/2021 (sesión 6 en el programa del evento, disponible el sitio web).

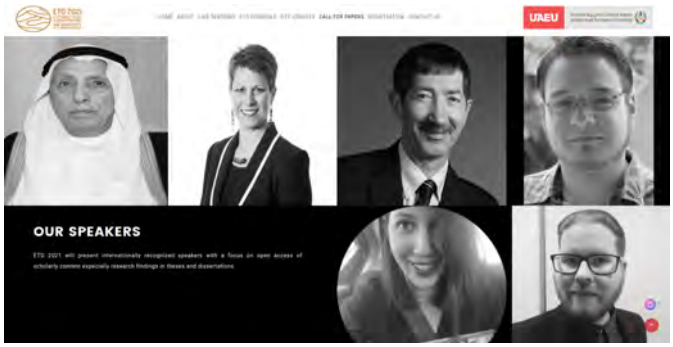


Figura 6.3: Captura de pantalla del sitio del ETD 2021

- Participación en el desafío ICON2021 Shared Task on Multilingual Gender Biased and Communal Language Identification (ComMA@ICON[2]) como parte del equipo LUC. ver figura 6.4

[1] https://conferences.uaeu.ac.ae/etd2021/en/index.shtml
[2] https://competitions.codalab.org/competitions/35482#results

Figura 6.4: Participación ICOM 2021

- Participación en el Foro Internacional de Evaluación Documental “Evaluación de documentos en la era digital” con la mesa redonda “Proyecto de datos personales e inteligencia artificial” en San José Costa Rica. ver figura 6.5

Figura 6.5: Constancia FIED 2023

6.1.3. Publicaciones

- Cuéllar-Hidalgo, R., Guerrero-Zambrano, J., Forest, D., Reyes-Salgado, G., & Torres-Moreno, J. (2021). LUC at ComMA-2021 Shared Task: Multilingual Gender Biased and Communal Language Identification without using linguistic feature, Disponible en: https://arxiv.org/abs/2112.10189

- Cuéllar-Hidalgo, R., Reyes-Salgado, G., & Torres-Moreno, J. M., (2022) Automatic Reference Mining: Review and perspectives. TextMine’22.
Disponible en: http://vincentlemaire-labs.fr/TM2022/ActesTextMine22.pdf (página 27)

- Barnard Amozurrutia, A., Bernal Astorga, Y., Cuéllar Hidalgo, R., Escoto Velázquez, C. A., & García-Velázquez, L. M., (2023). Inteligencia artificial en los archivos: Consideraciones de diseño e implementación. Tábula, (25), 41–59. https://doi.org/10.51598/tab.936
- Cuéllar Hidalgo, R., Pinto Elías, R., Torres-Moreno, J. M., Vergara Villegas, O. O., Reyes Salgado, G., & Magadán Salazar, A. (2024). Neural Architecture Comparison for Bibliographic Reference Segmentation: An Empirical Study. Data, 9(5), 71. https://doi.org/10.3390/data9050071 ***(JCR)***
- Capitulo del libro dictaminado como aceptado (ver figura 6.6).

MTRO. RODRIGO CUELLAR HIDALGO
P R E S E N T E

Estimado Mtro. Rodrigo Cuellar Hidalgo, con relación al texto denominado "El Impacto Transformador de la IA en la práctica Archivística", del cual Usted es autor, por este medio le informo que el texto presentado fue dictaminado por el Comité Editorial del INAI como Aprobado.

En ese sentido, con fundamento en lo dispuesto en el artículo 15, fracción VIII y 28 del Reglamento de Organización y funcionamiento del Comité Editorial del Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales (INAI), me permito notificarle el dictamen integral de mérito, el cual contiene algunas sugerencias al texto, mismas que se ponen a su amable consideración y valoración, para que, de estimarlas pertinentes, sean atendidas.

Para mayor referencia sobre el dictamen, le envío en archivo anexo al texto de su autoría que fue entregado a los integrantes del Comité Editorial.

Del mismo modo, con la finalidad de conocer su opinión y valoración respecto de las sugerencias, planteadas por el Comité Editorial, anexo al presente le envío a Usted el formato *Retroalimentación del Programa Editorial*, mismo que agradeceré remitirlo con el texto definitivo a esta Secretaría Técnica.

Las referidas sugerencias podrán integrarse y ser entregadas a más tardar el 8 de enero de 2024. Quedo a sus órdenes en el número telefónico 50042400 extensión 2475 y con el correo electrónico cristobal.robles@inai.org.mx.

Sin otro particular, reciba un cordial saludo.

Figura 6.6: Captura de pantalla Dictamen INAI

6.1.4. Otras actividades

- Revisor académico, de 3 artículos, Revista Komputer Sapiens, año15 volumen II, Sociedad Mexicana de Inteligencia Artificial (SMIA), ISSN: 2007-0691, ver figura 6.7
- Miembro del grupo de Trabajo Reference and Access en el proyecto InterPARES Trust AI[3] (2021-2026) ver figura 6.8

[3]https://interparestrustai.org/trust

SOCIEDAD MEXICANA DE INTELIGENCIA ARTIFICIAL
REVISTA KOMPUTER SAPIENS

ÍNDICE DE REVISTAS MEXICANAS
CONACYT DE DIVULGACIÓN CIENTÍFICA Y TECNOLÓGICA

CONSTANCIA DE REVISIÓN ACADÉMICA

México. D.F, 15 de mayo de 2023

Rodrigo Cuéllar Hidalgo

A nombre de la Sociedad Mexicana de Inteligencia Artificial (SMIA) le agradecemos su revisión académica de *3 artículos*, correspondiente al año 15 volumen II, del periodo mayo-agosto de 2023 de la revista de Komputer Sapiens, cuya temática central fue:

Tema libre

Agradecemos su colaboración con la revista *Komputer Sapiens* y lo invitamos a seguir participando con la misma.

Atentamente

Dra. Karina Mariela Figueroa Mora
Editora en jefe

Revista Komputer Sapiens ISSN: 2007-0691
Sociedad Méxicana de Inteligencia Artificial, A.C. Nuevo León 125, 303, Col. Hipódromo de la Condesa, C.P. 06140, DF, México.
Tel. +52 (833) 357.48.20 ext. 3024, fax +52 (833) 215.85.44
komputersapiens@smia.org.mx www.komputersapiens.org

Figura 6.7: Constancia de Revisión Académica SMIA.

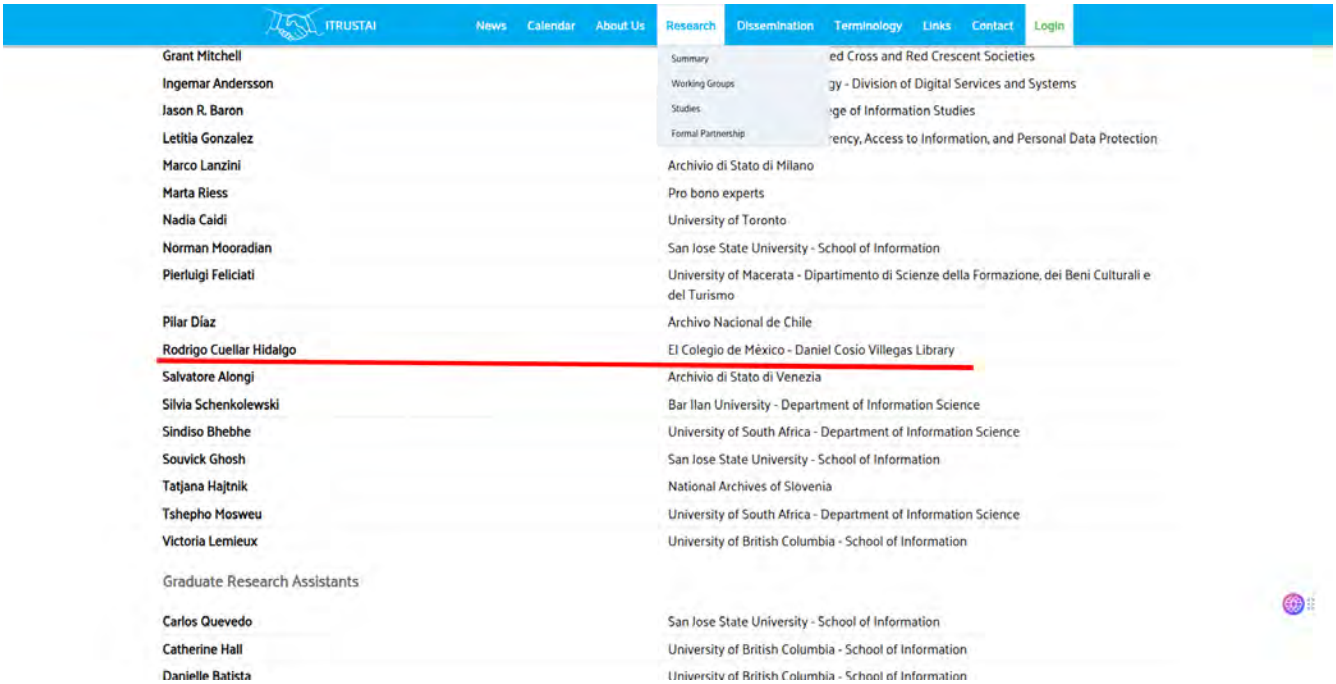


Figura 6.8: Captura de pantalla sitio InterPARES Trust IA