Análisis y selección automática de candidatos mediante lógica borrosa, procesamiento de lenguaje natural y aprendizaje profundo

Automatic analysis and selection of candidates using fuzzy logic, natural language processing, and deep learning

anonimizado

Resumen: En el contexto actual del proceso de selección de personal, la evaluación y elección del currículum adecuado para una oferta laboral enfrenta múltiples desafíos, como la abundancia de información no relevante en los currículos y la necesidad de que los departamentos de recursos humanos puedan justificar sus decisiones de selección ante posibles auditorías futuras. En este trabajo se presenta un procedimiento novedoso que, mediante reglas heurísticas y lógica borrosa, normaliza los diferentes currículos para su posterior análisis a través de un sistema que combina con éxito técnicas de procesamiento del lenguaje natural y aprendizaje profundo. La metodología desarrollada permite identificar con precisión los currículos que mejor se ajustan a una oferta laboral específica, dentro del sector tecnológico, y los criterios y reglas heurísticas utilizados y definidos por expertos en recursos humanos proporcionan a la decisión adoptada la fundamentación, transparencia y rigor necesarios.
Palabras clave: Lógica borrosa, Redes neuronales profundas, Toma de decisiones, Procesamiento de lenguaje natural.

Abstract: In the current context of the personnel selection process, the evaluation and choice of the appropriate resume for a job offer faces multiple challenges, such as the abundance of irrelevant information in resumes and the need for human resources departments to justify their selection decisions in the face of possible future audits. This work presents a novel procedure that uses heuristic rules and fuzzy logic to standardize different resumes for subsequent analysis through a system that successfully combines natural language processing and deep learning techniques. The developed methodology allows for accurately identifying the resumes that best match a specific job offer within the technology sector. Moreover, the criteria and the heuristic rules used and defined by human resources experts provide the decision with the necessary foundation, transparency, and rigor.
Keywords: Fuzzy logic, Deep neural networks, Decision making, Natural language processing.

1 Introducción

Los métodos tradicionales de selección de currículos resultan insuficientes para responder a las necesidades actuales de las empresas, que enfrentan un volumen masivo y diverso de datos de reclutamiento distribuidos en múltiples formatos y plataformas profesionales. La tarea de diferenciar adecuadamente entre palabras clave relacionadas con proyectos y habilidades específicas dentro de los currículums es compleja y consume un tiempo considerable. Además, en el contexto organizacional actual, la trazabilidad en el proceso de selección se ha convertido en un requisito esencial, demandando que los sistemas de selección faciliten la justificación futura de las decisiones de descarte de candidatos, garantizando

transparencia y responsabilidad en la gestión del talento.

El interés por optimizar la selección de currículums y eliminar el sesgo inherente a toda decisión tomada por el ser humano es evidente a tenor de los numerosos estudios aparecidos en los últimos veinte años. La clasificación de palabras mediante la agrupación de términos relacionados, y la estructuración y control del vocabulario específico de los currículums mediante el uso de tesauros fue propuesta por Kilgarriff (2003). Una de las soluciones para el reconocimiento de palabras clave en currículos que utiliza múltiples modelos lingüísticos fue propuesta por Wosiak (2021), quien implementó diccionarios específicos del idioma y dependencias lingüísticas complejas para el polaco. Por otra parte, diversos estudios como el realizado por Capiluppi, Serebreniky y Singer (2013), evidencian que las redes sociales profesionales contienen información pública de gran valor que permite identificar potenciales candidatos, incluso en ausencia de formación universitaria formal o credenciales convencionales. La obtención y el análisis de dicha información presenta dificultades añadidas vinculadas a este tipo de selección, aunque hoy en día existan motores de búsqueda sofisticados que operan directamente sobre estas plataformas. Esta tendencia subraya la importancia de manejar formatos de currículum provenientes de plataformas sociales profesionales tales como LinkedIn, X (antes Twitter), GitHub o Xing.

Las mejoras en la interpretación y clasificación de currículums se han apoyado históricamente en el uso de técnicas tradicionales de minería de datos (Giri et al., 2016; Bharadwaj et al., 2023), cuyo propósito es extraer términos clave y así superar las limitaciones inherentes al análisis manual por parte de evaluadores. La aplicación de técnicas de aprendizaje automático y procesamiento de lenguaje natural (PLN) también ha sido ampliamente descrita en la literatura (Thomas y Sangeetha, 2020, Harsha et al., 2022; Tayal et al., 2024). Más recientemente, se ha buscado la integración de técnicas de PLN y aprendizaje profundo (AP) (Sidi Boubacar y Ouardi, 2024), redes neuronales recurrentes (Jayadharshini et al, 2024) y transformadores (Bondielli y Marcelloni, 2021), para la mejora de la coincidencia entre las habilidades requeridas en los puestos de trabajo y los currículums de los candidatos. Por último, conviene destacar también el uso de la lógica borrosa o difusa (*fuzzy logic* en terminología inglesa) para la selección de personal, teniendo en cuenta factores como los rasgos de personalidad y la subjetividad (Kuppusamy et al., 2024), y permitiendo eliminar ambigüedades e incertidumbres semánticas (Xie y Zhao, 2025).

El uso combinado de estas técnicas podría constituir una estrategia prometedora para superar las limitaciones inherentes a los sistemas actuales de seguimiento de candidatos (*Applicant Tracking System*, ATS), tales como la imprecisión en la coincidencia de perfiles y la presencia de sesgos, optimizando así la eficacia del proceso de selección.

Las principales aportaciones de la propuesta presentada se sustentan, en primer lugar, en un sistema híbrido que integra un filtro de reglas heurísticas y lógica borrosa con técnicas tradicionales de aprendizaje automático para la detección y exclusión de currículums que no satisfacen los criterios predefinidos por expertos en recursos humanos. La principal ventaja de esta aproximación radica en su capacidad para ofrecer una parametrización rápida y una auditoría transparente durante la fase inicial de preselección, facilitando la supervisión y ajuste del proceso. Posteriormente, mediante una combinación de técnicas de PLN y AP, el sistema identifica con precisión los currículos que mejor se ajustan a una oferta laboral específica.

La organización del resto del documento es la siguiente: la segunda sección describe el proceso de normalización de currículums y la generación del *dataset*, y la propuesta presentada en este trabajo, en la que se describen las reglas heurísticas, los conjuntos difusos y las técnicas clásicas de aprendizaje automático utilizadas, así como las técnicas de PLN y AP que integran el enfoque combinado; la tercera sección presenta el análisis y discusión de los resultados obtenidos; y la cuarta sección expone las principales conclusiones e identifica las líneas de investigación futuras.

2 Solución propuesta

2.1 Normalización del dataset

El conjunto de datos utilizado en este trabajo está formado por 1050 currículums, recopilados a partir de perfiles profesionales disponibles públicamente en LinkedIn, Indeed, Xing y Tecnoempleo. Con el fin de respetar la

privacidad de los individuos, los datos han sido tratados de forma agregada y anonimizados, evitando el uso de información personal identificable.

Los currículums fueron recopilados realizando búsquedas específicas asociadas a cada categoría profesional, tales como «*Software Engineer*», «*Backend Developer*», «*Fullstack Developer*», «*QA Tester*», «*Software Testing*». A partir de los resultados obtenidos, se seleccionaron aquellos perfiles que cumplían los siguientes criterios: disponibilidad de información suficiente (experiencia, habilidades, etc.), coherencia entre el título profesional y las habilidades descritas, y asociación clara con una única categoría, tal como se muestra en la Tabla 1.

Categoría	Descripción
JP	Jefes de proyecto, managers y perfiles directivos en el ámbito tecnológico.
Arquitecto	Arquitectos de software, arquitectos de sistemas y administradores de bases de datos (DBA)
Analista	Analistas funcionales, consultores y perfiles técnicos de análisis
Programador senior	Desarrolladores con experiencia (fullstack o especializados)
Programador	Desarrolladores junior, perfiles en formación o con poca experiencia
Tester	Perfiles de aseguramiento de calidad (QA) y testing de software
Front	Desarrolladores especializados en frontend

Tabla 1: Categorías profesionales del conjunto de datos.

El proceso de recopilación se realizó de forma independiente para cada categoría con el objetivo de construir un *dataset* balanceado con 150 currículums por categoría. Posteriormente los currículums fueron normalizados de acuerdo con el estándar de facto, basado en el formato estructurado en archivos de exportación de LinkedIn. Este formato se caracteriza por una organización clara y coherente de la información en distintos archivos: *Profile.csv*, *Education.csv*, *Skills.csv*, *Certifications.csv*, *Positions.csv* y *Category.csv*, facilitando la estandarización para su análisis automatizado.

Para la adaptación estructural y estandarización de las diversas secciones presentes en cada currículum al formato de LinkedIn, se desarrolló una aplicación la cual admite como parámetros de entrada los archivos en formato HTML de las distintas plataformas de selección de personal. La salida del proceso de normalización genera una carpeta por cada currículum en cuyo interior se ubican los archivos CSV correspondientes. Estos archivos se utilizan como entrada normalizada en la siguiente fase del proceso. La Figura 1 muestra el diagrama del proceso de normalización.

2.2 Reglas heurísticas difusas y preprocesamiento

La propuesta presentada en este trabajo integra un filtro de preselección realizado con reglas heurísticas y conjuntos difusos. Gracias a este filtro, antes de realizar el proceso de coincidencia con la oferta laboral, es decir, de la clasificación del currículum, es posible que un currículum sea descartado. Un currículum descartado será aquel que a pesar de contar con palabras clave válidas acorde a los sistemas actuales ATS, no cuenta con los criterios heurísticos adecuados para dicha oferta. De este modo, solo aquellos currículums que cumplen con estos requisitos avanzan hacia el módulo de PLN y AP, como se representa en la Figura 1.

Para la realización de este filtro se han definido 5 criterios (en adelante, *Ci*) y 16 reglas heurísticas desarrolladas con la ayuda de expertos en recursos humanos. Los currículums normalizados sirven como entrada al sistema de control difuso, estableciendo así una analogía multicriterio para cada caso de entrada, emulando el criterio de evaluación empleado por los expertos, lo que facilita la generación de diversos valores de evaluación para cada candidato en función de cada criterio aplicado.

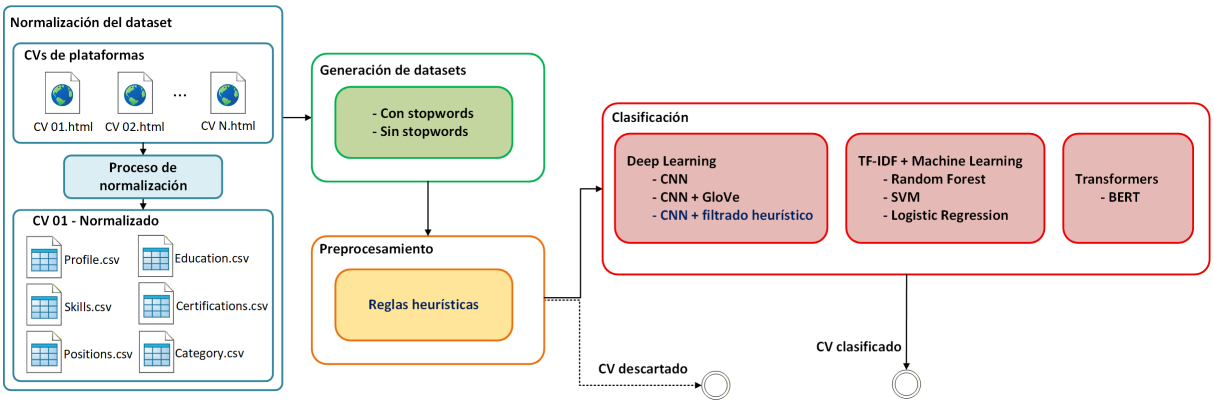

Figura 1: Flujo completo del sistema propuesto, desde la normalización de los currículums hasta la clasificación y evaluación de los modelos. Se generan dos variantes del *dataset* (con y sin *stopwords*), y se incorpora un módulo de filtrado heurístico, que constituye la principal contribución del trabajo. Se evalúan distintos enfoques, incluyendo modelos basados en TF-IDF y aprendizaje automático (*Machine Learning*), arquitecturas de aprendizaje profundo (*Deep Learning*) y modelos basados en transformadores (*Transformers*), con el objetivo de comparar su rendimiento.

A las puntuaciones obtenidas en cada criterio se les aplica distintas fórmulas de pertenencia a conjuntos difusos, lo que permite determinar si el candidato continúa o no en el proceso de contratación. No obstante, es posible continuar en el proceso aun no habiendo obtenido las puntuaciones necesarias, en cuyo caso se proporciona esta información como complemento para el personal de recursos humanos de cara a tomar o justificar la decisión futura de contratación.

La realización de entrevistas a expertos en recursos humanos fue la técnica empleada para definir y consensuar los criterios de este estudio y sus conjuntos difusos asociados (*anonimizado*):

- **Medición de la fiabilidad** (Criterio *CiValidez*, CV). Este criterio está enfocado en valorar la coherencia entre el puesto actual del candidato y la información reflejada en su currículum. Se examinan aspectos como la duración en el cargo, la inclusión de habilidades pertinentes y la formación requerida para el desempeño del puesto, entre otros indicadores clave que permiten evaluar la adecuación y preparación del candidato para el rol ofertado. Todos los criterios se han definido en términos de conjunto difuso en función de los cuantificadores bajo, medio y alto, en lenguaje FCL (Código 1).

```
FUZZIFY CiValidez
    TERM bajo:=(0,1)(0.1,1)(0.3,1)(0.4, 0);
    TERM medio := (0.3, 0) (0.5,1) (0.7,0);
    TERM alto := (0.6, 0) (0.7,1) (1, 1);
END_FUZZIFY
```

Código 1: Definición difusa para CiValidez.

- **Medición del compromiso** (Criterio *CiCompromiso*, CC). Este criterio mide el grado de compromiso y el tiempo continuado en cada empresa. Es crítico en ofertas con periodo de formación previo, ya que reduce la salida de candidatos tras formaciones específicas. Además, aporta información sobre rotación, ayudando a seleccionar perfiles con mayor permanencia. El criterio evalúa también aspectos como periodos de prueba y compromiso (Código 2).

```
FUZZIFY CiCompromiso
    TERM bajo := (0, 1)(0.1, 1)(0.3, 1) (0.4, 0);
    TERM medio := (0.3, 0) (0.5,1) (0.7,0);
    TERM alto := (0.5, 0)(0.8,1) (1, 1);
END_FUZZIFY
```

Código 2: Definición difusa para CiCompromiso.

- **Medición de la trayectoria** (Criterio *CiTrayectoria*, CT). Es importante para el empleador verificar la coherencia de la trayectoria profesional del currículum, asegurando que las habilidades estén alineadas con el puesto. Este criterio

analiza la progresión laboral, los tiempos entre etapas y la consistencia de la información, valorando evolución o retroceso (Código 3).

```
FUZZIFY CiTrayectoria
    TERM bajo := (0, 1)(0.1, 1)(0.3, 1) (0.4, 0);
    TERM medio := (0.3, 0) (0.4,1) (0.5,0);
    TERM alto := (0.5, 0) (0.7,1) (1, 1);
END_FUZZIFY
```

Código 3: Definición difusa para CiTrayectoria.

- **Medición de los conocimientos** (Criterio *CiConocimientos*, CCn). Valora la autenticidad y la relevancia de los conocimientos y las competencias. Permite detectar si el postulante ha incluido términos y expresiones clave únicamente con la finalidad de aumentar su visibilidad en plataformas de búsqueda de empleo, o si estos elementos son utilizados de forma adecuada, coherente y congruente con su perfil profesional (Código 4).

```
FUZZIFY CiConocimiento
    TERM bajo := (0, 1)(0.1, 1)(0.3, 1) (0.4, 0);
    TERM medio := (0.3, 0) (0.5,1) (0.7,0);
    TERM alto := (0.6, 0) (0.7,1) (1, 1);
END_FUZZIFY
```

Código 4: Definición difusa para CiConocimiento.

- **Medición global** (Criterio *CiGlobal*, CG). Este valor se obtiene calculando una media ponderada de los valores determinados para los criterios previos, proporcionando así una perspectiva completa y unificada del perfil del aspirante. La ponderación es parametrizable por el empleador mediante unos pesos, en función de las necesidades del puesto requerido (Código 5).

```
FUZZIFY CiGlobal
    TERM bajo := (0, 1)(0.1, 1)(0.3, 1) (0.4, 0);
    TERM medio := (0.3, 0) (0.5,1) (0.7,0);
    TERM alto := (0.6, 0) (0.7,1) (1, 1);
END_FUZZIFY
```

Código 5: Definición difusa para CiGlobal.

La aceptación de un currículum se basa así en criterios previamente definidos, garantizando una evaluación uniforme. Cada criterio se modela mediante funciones de pertenencia en un sistema difuso, diseñadas a partir de reglas heurísticas expertas. Este enfoque permite manejar la incertidumbre y obtener un valor que refleja el grado de cumplimiento de los requisitos.

Finalmente, se utiliza un conjunto difuso denominado *nivelAceptación*, que emplea el método del centroide o centro de gravedad para integrar todos los valores obtenidos y calcular un valor final de decisión. Este valor determina la idoneidad del currículum para continuar avanzando en el proceso de selección, asegurando una valoración integral y equilibrada del candidato. Este conjunto también está definido en función de los cuantificadores bajo, medio y alto, en lenguaje FCL (Código 6).

```
DEFUZZIFY nivelAceptacion
    TERM bajo := (0, 1)(0.1, 1)(0.3,1) (0.4,0);
    TERM medio := (0.3, 0) (0.5,1) (0.7,0);
    TERM alto := (0.6, 0) (0.7,1)(1, 1);
    METHOD : COG;
    DEFAULT := 0;
END_DEFUZZIFY
```

Código 6: Definición difusa para nivelAceptación.

En el filtro se define la contribución al consenso individual (CTC) para un criterio *i* como un valor entre 0 y 1 tal y como se describe en la Tabla 2.

CTC(*Ci*) ∈ [0,1]	Nivel de información
< 0,6	Currículum con información insuficiente o no veraz.
= 0,6	Currículum con información fiable y nivel aceptable de calidad.
> 0,6	Currículum con información muy precisa, fiable y veraz.

Tabla 2: Valores de CTC(*Ci*) para la búsqueda del consenso.

En el Código 7 se presenta, a modo de ejemplo, una reducida muestra de las reglas

heurísticas definidas por expertos en recursos humanos para su aplicación en el sistema.[1]

RULE 1: **IF** CiCompromiso **IS** bajo **AND** CiConocimiento **IS** bajo **AND** CiTrayectoria **IS** bajo **AND** CiValidez **IS** bajo **AND** CiGlobal **IS** bajo **THEN** nivelAceptacion **IS** bajo;

RULE 2: **IF** CiCompromiso **IS** medio **AND** (CiConocimiento **IS** medio **OR** CiConocimiento **IS** alto) **AND** (CiTrayectoria **IS** medio **OR** CiTrayectoria **IS** alto) **AND** (CiValidez **IS** medio **OR** CiValidez **IS** alto) **AND** (CiGlobal **IS** medio **OR** CiGlobal **IS** alto) **THEN** nivelAceptacion **IS** medio;

Código 7: Reglas heurísticas en lenguaje FCL.

Para verificar la coherencia junto a la efectividad del filtro y los criterios definidos, se realizó un análisis utilizando el método del Proceso de Análisis Jerárquico (AHP) (Saaty, 1987).

Uno de los principales problemas en la actualidad es que en los currículos se mencionan palabras clave y tecnologías relacionadas con los proyectos en los que ha trabajado un candidato, aunque dicho candidato no posea esas habilidades. En este sentido, la contribución clave de este trabajo, y que supone una mejora respecto a estudios previos, radica precisamente en el filtro descrito en esta sección y que, mediante técnicas heurísticas, realiza una validación de los currículums descartando los que no se ajustan a los criterios preestablecidos por expertos, a pesar de presentar una correspondencia a nivel de palabras clave. Por otro lado, las puntuaciones obtenidas por los candidatos en los criterios son visibles para los reclutadores con el fin de garantizar una trazabilidad y transparencia durante el proceso de selección automática. De este modo, una vez aplicado el filtro, los reclutadores pueden seleccionar un subconjunto o el conjunto completo de los currículums como entrada a la fase de PLN.

Para la representación de los currículums se empleó la técnica TF-IDF (*Term Frequency–Inverse Document Frequency*), ampliamente utilizada en tareas de clasificación de texto (Manning, Raghavan y Schütze, 2008). Esta técnica permite transformar cada currículum en un vector numérico ponderado en función de la relevancia de los términos dentro del corpus, reduciendo el impacto de palabras frecuentes poco informativas. Sobre estas representaciones vectoriales, se evaluaron distintos algoritmos de clasificación supervisada con el objetivo de establecer una línea base robusta frente a enfoques más complejos basados en redes neuronales. En primer lugar, se empleó un modelo de regresión logística, ampliamente utilizado en problemas de clasificación de texto debido a su eficiencia y capacidad de generalización en espacios de alta dimensionalidad (Manning y Schütze, 1999). Posteriormente se evaluó el rendimiento de máquinas de vector soporte (SVM), un algoritmo que ha demostrado un comportamiento competitivo en tareas de clasificación textual, especialmente cuando se combina con representaciones TF-IDF (Joachims, 1998), al maximizar el margen entre clases en espacios vectoriales de alta dimensión. Por último, se incluyó un modelo basado en Bosque aleatorio (*Random Forest*), con el objetivo de analizar el comportamiento de técnicas de ensamblado frente a modelos lineales, evaluando su capacidad para capturar relaciones no lineales en los datos. Estos modelos fueron evaluados bajo las mismas condiciones experimentales con el fin de realizar una comparación sistemática entre enfoques clásicos de aprendizaje automático. Los resultados obtenidos permiten establecer una referencia cuantitativa que será utilizada posteriormente para contrastar el rendimiento de los modelos basados en AP. Esta comparación permite además analizar las limitaciones de los enfoques tradicionales basados en bolsas de palabras frente a métodos que incorporan información contextual, como los modelos basados en transformadores y redes neuronales convolucionales descritos en las siguientes secciones.

2.3 Procesamiento del lenguaje natural

La siguiente fase del sistema consiste en un módulo de PLN destinado a transformar el conjunto de currículums vitae en representaciones numéricas aptas para su utilización en modelos de AP. En concreto, se propone un modelo basado en redes neuronales convolucionales (CNN) y un modelo basado en transformadores: BERT.

[1] El código fuente desarrollado se encuentra disponible en https://github.com/cnn-nlp-lab/sepln-paper-resources.

Es importante destacar que el preprocesamiento aplicado difiere en función del modelo utilizado. Mientras que la red CNN emplea una tokenización basada en índices y *embeddings* entrenables, el modelo BERT utiliza su propio tokenizador basado en subpalabras y *embeddings* contextuales preentrenados.

2.3.1 Representación secuencial CNN

El procedimiento se inicia con la carga de los currículums estandarizados tal y como se detalló en la sección 2.1 y consta de las siguientes fases:

- **Preprocesamiento**: Esta fase se centra en la limpieza del texto mediante la eliminación de elementos redundantes para reducir el ruido lingüístico. Para ello, se emplean herramientas como la biblioteca *Beautiful Soup* de Python (Richardson, 2024) y expresiones regulares que mejoran la calidad del corpus. El proceso incluye la eliminación de enlaces web, normalización de espacios, eliminación de tokens nulos y conversión a minúsculas. También se estandarizan las categorías para unificar etiquetas. Se generaron dos variantes del *dataset* para evaluar el preprocesamiento: una que mantiene las palabras comunes (*stopwords*), preservando el contexto para modelos basados en transformadores como BERT, y otra que las elimina para algoritmos clásicos de aprendizaje máquina, donde la reducción de ruido y dimensionalidad mejora el rendimiento (Manning, Raghavan y Schütze, 2008).
- **Tokenización**: En esta etapa, los textos son transformados en secuencias de índices enteros mediante el uso de un tokenizador. A diferencia de las representaciones basadas en frecuencias como TF-IDF, este enfoque asigna un identificador único a cada término del vocabulario, permitiendo preservar el orden secuencial de las palabras en el documento. En este trabajo se emplea el tokenizador proporcionado por Keras, configurado con un tamaño máximo de vocabulario. Este enfoque resulta adecuado para el tratamiento de currículums, donde aparecen términos técnicos, siglas y expresiones con alta variabilidad. Además, se incluye un token para palabras desconocidas (*out-of-vocabulary*), lo que mejora la robustez del modelo frente a términos no vistos durante el entrenamiento.
- **Generación de *Embeddings***: A partir de las secuencias tokenizadas, los términos son proyectados a vectores densos mediante una capa de *embedding* entrenable dentro de la red neuronal. Esta representación permite capturar relaciones semánticas entre palabras durante el proceso de entrenamiento.
- **Aplicación de *Padding***: Para garantizar la homogeneidad de las secuencias de entrada, los vectores con una longitud menor son completados con valores nulos (ceros) de manera que todas las secuencias representarán valores uniformes, asegurando así su compatibilidad con los modelos de AP. Esta técnica de *padding* es fundamental para el procesamiento adecuado en arquitecturas que requieren entradas con el mismo tamaño.

La salida generada por este proceso compuesta por las secuencias tokenizadas, las etiquetas y el tamaño del vocabulario se utiliza como entrada para el entrenamiento de una red neuronal convolucional unidimensional multiescala siguiendo el enfoque propuesto en (*anonimizado*).

2.3.2 Modelo basado en transformadores

Para capturar información contextual, en concreto se utilizó BERT (*Bidirectional Encoder Representations from Transformers*), un modelo de lenguaje preentrenado basado en transformadores que ha demostrado un alto rendimiento en PLN (Devlin et al., 2019). A diferencia de las representaciones tradicionales como TF-IDF, BERT genera *embeddings* contextuales, lo que implica que la representación de una palabra depende del contexto en el que aparece, permitiendo capturar relaciones semánticas más complejas.

Para la implementación, los textos fueron previamente tokenizados utilizando el tokenizador asociado a BERT, respetando su esquema de subpalabras. Posteriormente, los

datos fueron introducidos en el modelo para su entrenamiento supervisado.

Tras un proceso de ajuste fino (*fine-tuning*), se evaluó el rendimiento del modelo comparándolo con los enfoques clásicos descritos anteriormente, con el objetivo de analizar la mejora aportada por el uso de representaciones contextuales. Este enfoque permite por tanto analizar el impacto del uso de modelos de lenguaje avanzados frente a técnicas tradicionales basadas en frecuencias de término, y frente a la arquitectura basada en redes CNN descrita en la siguiente sección.

2.4 Clasificación mediante una red neuronal convolucional

La red neuronal implementada en esta fase es de tipo convolucional, diseñada para clasificación multiclase de textos como los currículums. En la capa de *embedding* se evaluaron dos enfoques: *embeddings* entrenados desde cero durante el proceso de aprendizaje y *embeddings* preentrenados basados en GloVe (*Global Vectors for Word Representation*) (Pennington, Socher y Manning, 2014). El uso de *embeddings* preentrenados permite incorporar conocimiento semántico previo obtenido a partir de grandes corpus textuales, lo que puede resultar especialmente beneficioso en escenarios con conjuntos de datos limitados. Por otro lado, los *embeddings* entrenados desde cero permiten adaptar completamente las representaciones al dominio específico del problema.

La red utiliza tres familias de filtros convolucionales 1D con diferentes tamaños de *kernel* (2, 3 y 4), que capturan patrones locales de bigramas, trigramas y cuatrigramas respectivamente. Cada conjunto de filtros genera 50 mapas de características, aplicando la función de activación ReLU (Unidad Lineal Rectificada) para introducir no linealidad.

Cada salida convolucional pasa por una capa de *pooling*, utilizando la función *GlobalMaxPool1D*, que resume la información más relevante de cada característica a un valor máximo, reduciendo la dimensionalidad y preservando las características más significativas. Las salidas son concatenadas para su clasificación en una capa densa de 512 neuronas y activación ReLU. A continuación, la red incorpora una capa de *dropout* para prevenir el sobreajuste durante el entrenamiento. Por último, una capa densa con función de activación *softmax* genera las probabilidades para las clases posibles, y permite clasificar cada currículum en una categoría específica.

Con el objetivo de optimizar el rendimiento del modelo, se llevó a cabo una búsqueda sistemática de hiperparámetros mediante un enfoque de búsqueda en cuadrícula (*grid search*). Se exploraron distintas configuraciones de parámetros relevantes, incluyendo el número de filtros convolucionales, la tasa de *dropout*, el tamaño de lote (*batch s*ize) y la tasa de aprendizaje (*learning rate*). Asimismo, se consideraron diferentes escenarios experimentales, incluyendo la exclusión de determinadas clases y subconjuntos de datos, con el fin de analizar la robustez del modelo frente a variaciones en el conjunto de entrenamiento.

El modelo fue entrenado utilizando el optimizador Adam con distintas tasas de aprendizaje, minimizando la función de pérdida de entropía cruzada categórica (*sparse categorical crossentropy*). Se empleó la técnica de parada temprana (*early stopping*), monitorizando la pérdida en validación, para detener el entrenamiento cuando no se observara una mejora significativa y restaurando los mejores pesos obtenidos durante el entrenamiento.

3 Resultados

3.1 Configuración experimental y métricas de evaluación

Con el objetivo de evaluar el rendimiento de los distintos enfoques propuestos, se definió un protocolo experimental común para todos los modelos analizados.

La evaluación se realizó utilizando métricas estándar en tareas de clasificación multiclase, incluyendo la exactitud (*accuracy*) y el Valor-F (*F1-score*) (macro), este último especialmente relevante en presencia de posibles desbalances entre clases. Adicionalmente, se analizaron los resultados mediante matrices de confusión [2], lo que permitió identificar patrones de error y confusión entre categorías específicas. Asimismo, se consideraron métricas relacionadas con el coste computacional, incluyendo el tiempo de entrenamiento, el tiempo de inferencia y el consumo de memoria.

[2] Disponibles, por restricciones de espacio, en https://github.com/cnn-nlp-lab/sepln-paper-resources.

Estas medidas permitieron evaluar no solo la precisión de los modelos, sino también su viabilidad en entornos reales.

Todos los modelos fueron evaluados utilizando particiones consistentes de entrenamiento y test, con el fin de garantizar la comparabilidad de los resultados. El conjunto de datos fue dividido en dos subconjuntos: un 80% de los datos se destinó al entrenamiento, mientras que el 20% restante se utilizó para la evaluación final. Esta partición se realizó manteniendo la proporción de clases mediante un muestreo estratificado, garantizando así la representatividad de todas las categorías en ambos conjuntos. A partir de esta configuración experimental, se llevó a cabo el análisis comparativo descrito en la sección siguiente.

Modelo	Accuracy	F1-score	Entrenamiento (s)	Inferencia (ms/doc)	Memoria (MB)
TF-IDF + Regresión logística	**0,8048**	**0,8035**	0,08	0,003	**337,7**
TF-IDF + SVM	0,7857	0,7838	**0,02**	**0,001**	363,2
TF-IDF + Random Forest	**0,8048**	0,8014	0,63	0,045	396,4
BERT	0,7381	0,7355	3363,39	356,150	3093,6
CNN (mejor configuración)	0,7905	0,7902	22,71	0,570	2581,2

Tabla 3: Comparación entre los distintos modelos del estudio.

3.2 Resultados experimentales y análisis comparativo

En primer lugar, se realizó una comparación global entre los distintos enfoques evaluados, incluyendo modelos clásicos basados en TF-IDF, modelos de lenguaje preentrenados y redes neuronales profundas. Dicha comparación se ilustra en la Tabla 3.

Los resultados muestran que los modelos clásicos basados en TF-IDF, especialmente la regresión logística, ofrecen un rendimiento competitivo, alcanzando valores de F1-score superiores a 0,80. Por otro lado, el modelo basado en BERT presenta un rendimiento inferior en este caso, lo que puede deberse al tamaño limitado del *dataset* y al coste asociado al proceso de ajuste fino. La red CNN obtiene resultados comparables a los modelos clásicos, situándose como la mejor alternativa dentro de los enfoques de AP evaluados.

El coste computacional varía significativamente entre los modelos evaluados. Los modelos clásicos presentan tiempos de entrenamiento e inferencia prácticamente despreciables, lo que los hace adecuados para entornos con recursos limitados. Por el contrario, BERT presenta un coste computacional muy elevado, tanto en entrenamiento como en inferencia, lo que limita su aplicabilidad en escenarios reales sin recursos especializados. La red CNN ofrece un compromiso adecuado entre rendimiento y coste computacional, posicionándose como una alternativa equilibrada. Esto unido a lo anterior motivó su elección final. A partir de aquí, se analizaron diferentes escenarios con el fin de mejorar el rendimiento del modelo.

El uso de *embeddings* preentrenados basados en GloVe frente al entrenamiento desde cero, no mejoró el rendimiento del modelo en este caso, tal como se puede constatar en la Tabla 4. Esto sugiere que, dado el dominio específico de los currículums y la presencia de terminología técnica, resulta más efectivo aprender representaciones directamente a partir de los datos disponibles.

Configuración CNN	Accuracy	F1-score
CNN (embeddings propios)	**0,7905**	**0,7902**
CNN + GloVe	0,7476	0,7492

Tabla 4: Impacto de *embeddings* sobre el modelo CNN.

La búsqueda de hiperparámetros permitió identificar configuraciones óptimas del modelo (Tabla 5). Los resultados muestran que el mejor rendimiento se alcanza con un número intermedio de filtros (96), mientras que

incrementos adicionales no aportan mejoras significativas e implican un mayor coste computacional, especialmente en términos de memoria. Asimismo, valores intermedios de *dropout* (0,45–0,5) ofrecen un buen equilibrio entre generalización y capacidad de aprendizaje.

Filtros	Dropout	Accuracy	F1-score
96	0,5	**0,7905**	**0,7902**
96	0,45	0,7810	0,7804
128	0,45	0,7810	0,7792

Tabla 5: Mejores configuraciones del modelo CNN.

Con el objetivo de analizar la robustez del modelo, se evaluaron distintos escenarios mediante la exclusión de clases y subconjuntos de datos. Los resultados muestran una mejora significativa al eliminar determinadas clases, lo que sugiere la existencia de solapamiento semántico o mayor dificultad de clasificación en dichas categorías. En particular, la eliminación de las clases «*Analista*» y «*Programador senior*» produce el mayor incremento del rendimiento (Tabla 6), lo que indica que estas clases introducen ambigüedad en el proceso de clasificación.

Escenario	Accuracy	F1-score
Dataset completo	0,7905	0,7902
Sin clase «Analista»	0,8222	0,8220
Sin clase «Programador senior»	0,8444	0,8455
Sin clases «Analista» y «Programador senior»	**0,8733**	**0,8727**

Tabla 6: Resultados del modelo CNN tras la exclusión de clases y subconjuntos de datos.

Este análisis pone de manifiesto la importancia de la calidad y el filtro de las clases en tareas de clasificación de perfiles profesionales.

4 Conclusiones

En este trabajo se ha abordado la problemática de la clasificación automática de currículums mediante una evaluación comparativa de diferentes enfoques, incluyendo métodos clásicos basados en TF-IDF, modelos de lenguaje preentrenados y arquitecturas de aprendizaje profundo (CNN y transformadores).

Los resultados obtenidos muestran que los modelos tradicionales, especialmente la regresión logística, ofrecen un rendimiento competitivo con un coste computacional reducido, lo que los convierte en una opción eficiente en escenarios con recursos limitados. Por otro lado, los modelos basados en transformadores como BERT presentan un mayor coste computacional sin lograr mejoras significativas en el contexto del *dataset* utilizado. En cuanto a los modelos de aprendizaje profundo, la red neuronal convolucional propuesta alcanza resultados comparables a los mejores modelos clásicos, mostrando además un comportamiento robusto ante distintas configuraciones experimentales.

Uno de los principales aportes de este trabajo radica en el proceso de filtrado y selección de datos previo al entrenamiento, el cual permite mejorar la calidad del conjunto de datos y, en consecuencia, el rendimiento de los modelos. Asimismo, el estudio de ablación realizado ha puesto de manifiesto la existencia de solapamientos semánticos entre perfiles profesionales, lo que dificulta la separación entre ciertas categorías. Este hallazgo refuerza la importancia de un adecuado diseño del *dataset* en tareas de clasificación textual.

Finalmente, este trabajo destaca la necesidad de considerar no solo la precisión de los modelos, sino también su coste computacional y su aplicabilidad en entornos reales. Como líneas futuras, se plantea la ampliación del *dataset* y la posible liberación de una versión anonimizada teniendo en cuenta las consideraciones éticas asociadas al uso de información procedente de fuentes públicas [3], y la mejora de los criterios de filtrado valorando el grado de ajuste de un candidato a un puesto determinado teniendo en cuenta no solo sus competencias técnicas, sino también otros aspectos que la organización considera estratégicos en la evaluación global del perfil.

Agradecimientos

anonimizado

Bibliografía

Bharadwaj, R., D. Mahajan, M. Bharsakle, K. Meshram, y H. Pujari. 2023. Resume

[3] No obstante, se describe cómo generar el *dataset* utilizado en este trabajo en https://github.com/cnn-nlp-lab/sepln-paper-resources

analysis using NLP. En *7th International Conference on Inventive Systems and Control (ICISC)*, páginas 551-561, Coimbatore, India.

Bondielli, A., y F. Marcelloni. 2021. On the use of summarization and transformer architectures for profiling résumés. *Expert Systems with Applications*, 184:115521.

Capiluppi, A., A. Serebrenik, y L. Singer. 2013. Assessing technical candidates on the social web. *IEEE Software*, 30(1):45-51.

Devlin, J., M.-W. Chang, K. Lee, y K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. En Proceedings of NAACL-HLT 2019 (páginas 4171–4186).

Giri, A., A. Ravikumar, S. Mote, y R. Bharadwaj. 2016. Vritthi - a theoretical framework for IT recruitment based on machine learning techniques applied over Twitter, LinkedIn, SPOJ and GitHub profiles. En *International Conference on Data Mining and Advanced Computing (SAPIENCE)*, páginas 1-7, Ernakulam, India.

Harsha, T.M., G.S. Moukthika, D.S. Sai, M.N.R. Pravallika, S. Anamalamudi, y M. Enduri. 2022. Automated resume screener using natural language processing (NLP). En *6th International Conference on Trends in Electronics and Informatics (ICOEI)*, páginas 1772-1777, Tirunelveli, India.

Jayadharshini, P., S. Karuppusamy, S. Santhiya, J. Rakshitaa, N. Abarna, y N. Kannan. 2024. Harnessing NLP to align Candidate Profile with Position Requirements. En *15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, páginas 1-7, Kamand, India.

Joachims, T. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026683.

Kilgarriff, A. 2003. Thesauruses for natural language processing. En *International Conference on Natural Language Processing and Knowledge Engineering*, páginas 5-13, Beijing, China.

Kuppusamy, M., J. Singh, P. Raman, y N. Abdolrahimi. 2024. Advanced decision analytics with fuzzy logic integrating AI and computational thinking for personnel selection. *Journal of Fuzzy Extension and Applications*, 5(4):679-699.

Manning, C. D., P. Raghavan, y H. Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press.

Manning, C. D., y H. Schütze. 1999. Foundations of statistical natural language processing. MIT Press.

Pennington, J., R. Socher, y C. D. Manning. 2014. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (páginas 1532–1543). Association for Computational Linguistics.

Richardson, L. 2024. Beautiful Soup. https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Saaty, R.W. 1987. The analytic hierarchy process—what it is and how it is used. *Mathematical Modelling*, 9:161-176.

Sidi Boubacar, Z. y F. Ouardi. 2024. Automating Recruitment Process Using NLP and Deep Learning: A Novel Approach for Accurate Candidate Selection. En *World Conference on Complex Systems (WCCS)*, páginas 1-8, Mohammedia, Morocco.

Tayal, S., T. Sharma, S. Singhal, y A. Thakur. 2024. Resume screening using machine learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10:602-606.

Thomas, A., y S. Sangeetha. 2020. Intelligent sense-enabled lexical search on text documents. En *Proceedings of the 2019 Intelligent Systems Conference (IntelliSys)*, vol. 2, páginas 405-415, London, UK.

Wosiak, A. 2021. Automated extraction of information from Polish resume documents in the IT recruitment process. *Procedia Computer Science*, 192:2432-2439.

Xie, J., y L. Zhao. 2025. Research on resume screening and career matching model based on fuzzy natural language processing.

Journal of Computational Methods in Sciences and Engineering.