Metodología para la clasificación de documentos de texto de hojas de vida basado en aprendizaje de máquina

Estudiante:

Javier Leomar Matamoros Villegas

Director:

Edwin Nelson Montoya Munera

Departamento de Informática y Sistemas

emontoya@eafit.edu.co

Investigador Junior – Colciencias

Profesor-Investigador - EAFIT

Palabras Clave: Datos no Estructurados, Recursos Humanos, Aprendizaje Automático, Análisis de Texto, Text Analysis

TABLA DE CONTENIDO

Listado de Figuras

Listado de Tablas

Resumen

El proceso de selección de personal es complejo y requiere una gran cantidad de información y análisis para encontrar a los candidatos adecuados para una posición. Incluye varias etapas, como la revisión de currículums, pruebas psicológicas y verificación de referencias. Sin embargo, el análisis de currículums puede ser un desafío, ya que implica una intervención humana y la gran cantidad de información puede resultar difícil de procesar por computadora. Además, las empresas pueden enfrentar dificultades y costos elevados debido a la complejidad del proceso y la alta demanda en el mercado laboral.

Para resolver este problema, se propone la metodología CVNLP (Curriculum Natural Language Processing), que utiliza un conjunto de 725 hojas de vida en formatos PDF, DOCX y DOC para analizar los currículums de manera eficiente y eficaz. La metodología se aplica de manera transversal y ha demostrado su eficacia en la selección de personal. Al reducir los costos y mejorar la eficiencia en el proceso de selección de personal, las empresas pueden centrarse en su núcleo de negocio y facilitar el proceso de selección de personal. En resumen, la metodología CVNLP se presenta como una solución prometedora para mejorar la eficacia y eficiencia en los procesos de selección de personal, especialmente para las PYMEs con recursos limitados.

1. Introducción

El análisis de currículos es un proceso importante en la etapa de contratación de personal para las empresas, dada la importancia de detallar correctamente el perfil de los aspirantes [1]. Regularmente los procesos de selección de personal agrupan un número significativo de solicitantes para observar en detalle las habilidades de los candidatos, lo que genera grandes esfuerzos para seleccionar a la persona idónea para asumir un cargo. Dada la cantidad de información que debe ser observada, el uso de herramientas informáticas es importante, sin embargo, el análisis de los documentos en formato de texto puede dificultar el proceso, por esta razón, la comunidad científica apoya el uso de algoritmos de inteligencia artificial para el análisis de documentos no estructurados en texto, tal como se observa en [2] [3] [4].

1.1 Descripción del Problema

Los procesos de selección de personal tienen como principal objetivo, integrar a los equipos de trabajo el personal idóneo de acuerdo con un perfil solicitado. En el proceso de selección, se pueden encontrar diferentes etapas, tal como se expone en [5].

- Análisis del currículum.
- Verificación de requisitos.
- Pruebas psicológicas y/o técnicas.

- Verificación de referencias y experiencia laboral.
- Entrevistas.

Estas etapas generan que los procesos de selección de personal sean complejos dada la cantidad de información que debe ser analizada, lo que puede incurrir en costos para las empresas.

Teniendo en cuenta el crecimiento exponencial del mercado laboral, la cantidad de solicitudes por parte de interesados en puestos vacantes ha desbordado la capacidad humana y de las empresas para realizar procesos de selección objetivos, dada la gran cantidad de datos e información que deben ser analizados. Las empresas encargadas de los procesos de selección de personal se ven afectadas dado que no tienen la capacidad de realizar un seguimiento riguroso de las vacantes y de los candidatos.

Uno de los mayores retos en los procesos de selección de personal, es el análisis de los currículos enviados por los aspirantes a una vacante, dado que se hace necesaria la intervención humana para la lectura e interpretación de la información, sumado a la gran cantidad de archivos que se pueden recibir en una empresa en formatos *Portable Document Format* (PDF). Teniendo en cuenta que los archivos PDF no son datos estructurados procesables por computador que faciliten la extracción útil para un procesamiento automático.

Otro reto es el que se genera debido a la cambiante naturaleza de la información, que va desde la formación académica hasta formación personal, este tipo de información debe ser tenido en cuenta en el desarrollo, puesto que la manera de abordar la situación puede variar drásticamente.

Todos estos factores, generan dificultades para los encargados de los procesos de selección de personal, dada las implicaciones en términos de tiempo, costos en aplicación de filtros y rendimiento [6].

Para las PYMES, el análisis de los currículos puede generar grandes problemáticas, dado que no es posible que puedan acceder a los diferentes recursos que existen en el mercado, rezagándolas e impidiendo su crecimiento. Mencionado lo anterior, una herramienta de análisis (clasificación) de currículos, que sea de uso libre en el ámbito empresarial, permitiría crecer y dar una mejor respuesta, no solo a sus clientes externos a través de procesos de selección objetivos y rigurosos, sino con sus mismos empleados, o cliente interno, a través de la entrega de mejores herramientas, más adecuadas para cada tarea, lo cual se traduce en menores costos, mayor tiempo de respuesta, capacidad de atender más clientes y un crecimiento más constante.

Actualmente, la labor de extraer y clasificar la información de los currículos se hace manualmente, esta información se debe analizar diariamente y en muchas ocasiones, varias veces al día. De aquí surge la necesidad de tener una herramienta que facilite el trabajo a quien hace esta revisión sistemática de competencias, calificaciones y cualificaciones; facilitando así que la empresa se pueda enfocar en el núcleo del negocio que es el proceso de selección.

1.2 Objetivo General

Validar una metodología y diferentes modelos de aprendizaje de máquina para la clasificación automática de hojas de vida o currículos profesionales en formatos de texto como apoyo al análisis automático o semiautomático de la información para mejorar la toma de decisiones.

1.3 Objetivos Específicos

1. Recolección de los datos a partir del problema de investigación.
2. Recolección de un conjunto de datos sobre currículos profesionales en texto para realizar análisis exploratorio y posterior entrenamiento de modelos de clasificación basado en aprendizaje automático.
3. Seleccionar y evaluar diferentes técnicas y modelos de clasificación de documento en texto de currículos profesionales que mejor rendimiento demuestren en la extracción automática o semiautomática de información.
4. Aplicar la metodología propuesta con diferentes conjuntos de datos relacionados con currículos.

1.4 Descripción de los capítulos

El capítulo 1 presenta el contexto general acerca del análisis de CV usando técnicas de aprendizaje de máquina. De allí, se parte para la definición del problema de investigación y el aporte que se espera generar con el presente trabajo. Se define el objetivo general y se plantean los objetivos específicos.

El capítulo 2 presenta el marco teórico y estado del arte de diferentes modelos de aprendizaje de máquina, entendimiento natural del lenguaje, metodologías, entre otros.

El capítulo 3 presenta la adaptación de una metodología para el Análisis de Currículos con NLP (CVNLP). Allí se realiza una descripción de cada uno de los pasos de la metodología y se mencionan los pasos para la implementación de la propuesta.

El capítulo 4 presenta la aplicación de la metodología en función de lo establecido en el capítulo 3. Aquí se relacionan los resultados obtenidos con respecto a lo propuesto en el capítulo 3. Se resaltan los resultados obtenidos en cada etapa de la metodología.

El capítulo 5 presenta el análisis de los resultados en todo el proceso del proyecto y las conclusiones a las cuales se llegaron después del análisis de resultados. Adicional, se definen posibles trabajos futuros.

2 Marco teórico y Estado del arte

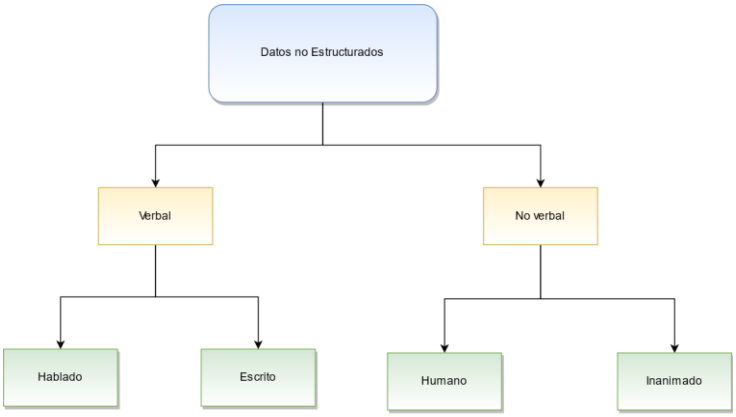
2.1 Marco teórico

A continuación, se relacionan los conceptos relevantes para este trabajo, teniendo en cuenta la revisión realizada en la comunidad académica. Se relacionan conceptos asociados a Datos no Estructurados, Aprendizaje Automático, Selección de Características y Modelos de Ensamble.

2.1.1 Datos no estructurados

Se denominan Datos no estructurados a aquellos que no se almacenan bajo un formato predefinido y en caso de poseer uno, no es fácilmente interpretable por aplicaciones computacionales [7]. En [8], se propone un esquema para representar los Datos No Estructurados:

Figura 1: Esquema de Datos no Estructurados

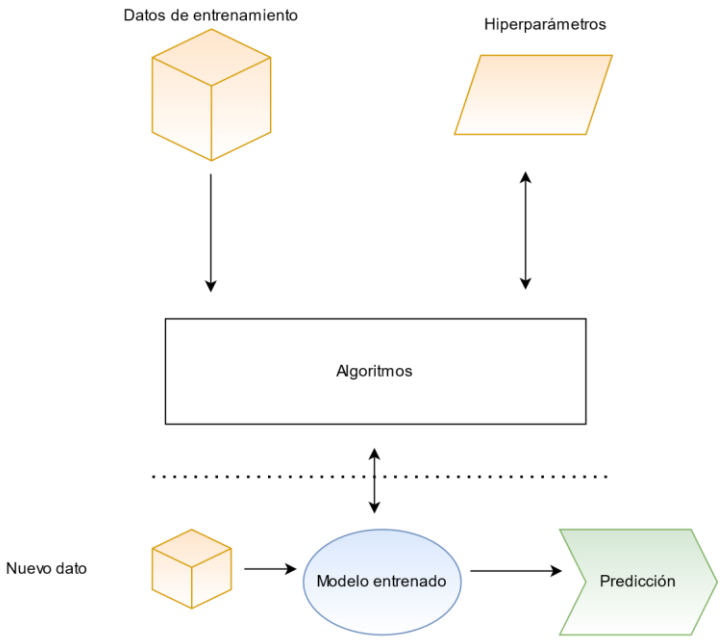


Tomado desde: [8]

2.1.2 Aprendizaje automático o Aprendizaje de máquina

El Aprendizaje Automático (ML o Machine Learning) se define como una rama de la inteligencia artificial que se soporta en modelos matemáticos, buscando que las máquinas puedan adaptarse de manera independiente a diversas situaciones [9].

Figura 2: Esquema de Aprendizaje Automático



Fuente: Tomado de [9].

2.1.2.1 Aprendizaje Supervisado

El Aprendizaje Supervisado se define como una estrategia que se soporta en el entrenamiento de diferentes algoritmos de Aprendizaje Automático cuyos datos se encuentran clasificados o categorizados mediante una etiqueta. Su principal característica, es la necesidad de datos etiquetados para la validación del modelo y para el aprendizaje [10].

A continuación, se describen algunas técnicas de clasificación supervisada:

- **Naïve Bayes**

El clasificador *Naïve Bayes* proporciona una buena línea de base para esta tarea utilizando la variante multinomial. Es un popular teorema de probabilidad de Bayes y el uso rápido de la técnica de aprendizaje automático para la clasificación de documentos.

$$P(C|f) = \frac{P(f|C) * P(C)}{P(f)}$$

El modelo probabilístico puede interpretarse como: "*cuál es la probabilidad de que el objeto pertenezca a la clase con ocurrencia de una característica en clase*".

P(f) indica probabilidad de que la característica *f* ocurra cuántas veces en la clase *C.*

P(f|C) Indica cuál es la probabilidad de ocurrencia del evento f dado el evento C.

- **Máquina de Vectores de Soporte o SVM (Support Vector Machine)**

En [11] el SVM se define como un modelo lineal para problemas de clasificación y regresión. Puede resolver problemas lineales y no lineales creando una línea o hiperplano que separa los datos en clases.

La clasificación de vectores de soporte lineal es como SVM con el parámetro kernel = 'lineal', pero implementado en términos de liblinear en lugar de libsvm, por lo que tiene más flexibilidad en la elección de penalizaciones y funciones de pérdida y debería escalar mejor a un gran número de muestras.

- **Regresión logística**

La regresión logística es un clasificador lineal que mide la relación entre la variable dependiente categórica y una o más variables independientes mediante la estimación de probabilidades utilizando una función logística/sigmoidea [12].

- **Bosque Aleatorio**

Los modelos de bosque aleatorio son un tipo de algoritmo usado en Aprendizaje de Máquina. Son parte de la familia de modelos basados en árboles. Desarrolla muchos árboles de decisiones con selección aleatoria de datos y estos árboles forman bosque. En el árbol de decisión, cada nodo hoja está dando diferentes predicciones sobre la clase y este método elige mejor entre el subconjunto de predictores para lograr una mejor precisión [13].

- **K Vecinos Cercanos (KNN)**

Algoritmo que permite clasificar datos en el grupo al que mejor corresponda, calculando la distancia entre un elemento nuevo y los anteriores. Ordena las distancias para determinar a qué grupo pertenece cada valor [14].

- **Descenso Gradiente Estocástico (*Stochastic Gradient Descent - SGD*)**

Descenso Gradiente Estocástico es un algoritmo iterativo que comienza desde un punto aleatorio en una función y viaja por su pendiente en pasos hasta que alcanza el punto más bajo de esa función. El descenso de gradiente estocástico determina dónde es posible inducir potencialmente la aleatoriedad en un algoritmo de descenso del gradiente.

- **Árboles de Decisión**

Este modelo predictivo emula el funcionamiento de un árbol, el cual está compuesto por nodos, vectores, flechas y etiquetas. Se puede definir como un mapa que permite generar posibles resultados o clasificaciones en función de una serie de decisiones o valores representados dentro de un modelo [15].

2.1.2.2 Aprendizaje Semi-Supervisado

Según lo expuesto en [16], el Aprendizaje Semi-Supervisado es una derivación del Aprendizaje Automático que se ocupa de analizar los datos con y sin etiquetas para realizar labores de clasificación. El Aprendizaje Semi-Supervisado (SSL), refleja un buen rendimiento en los modelos predictivos con el uso de datos no etiquetados y en particular, con la clasificación de imágenes [17].

Aprendizaje no Supervisado

El Aprendizaje no Supervisado es una estrategia que permite encontrar patrones en un conjunto de datos, aprender estructuras subyacentes, eliminar redundancia y limitar la dimensionalidad en grandes volúmenes de registros que no se encuentran etiquetados [18]. Su aplicación es variada, no obstante, suele combinarse con modelos generativos donde se debe modelar una característica que se extrae de un conjunto de datos original [19].

A continuación, se describen algunas de las técnicas implementadas en el presente trabajo.

- **Agrupación de K- Medias (K-Means Clustering)**

Agrupación de K-Medias es un método de cuantización vectorial, que tiene como objetivo dividir n observaciones en k clústeres en los que cada observación pertenece al clúster con la media más cercana (centros de clúster o centroide de clúster), sirviendo como prototipo del clúster. Esto da como resultado una partición del espacio de datos en celdas. Agrupación de K-Medias minimiza las varianzas dentro del clúster (distancias euclidianas al cuadrado).

2.1.3 Técnicas de Aprendizaje de Profundo

Las Técnicas Aprendizaje Profundo se pueden definir como un subconjunto derivado del Aprendizaje Automático que realiza tareas de clasificación, en la mayoría de los casos con un conjunto de datos grande, en comparación al Aprendizaje de Máquina [20].

Buscando mejorar las métricas obtenidas en las técnicas de Aprendizaje de Máquina, se implementaron algoritmos LSTM (*Long short-term memory*) y CNN (*Convolutional Neural Networks*), dados los buenos resultados obtenidos en [21].

Las Redes Neuronales Convolucionales son otra propuesta asociada al aprendizaje supervisado, donde inicialmente su uso se asociaba con el procesamiento de imágenes, no obstante, dado su estudio, se han encontrado otras características que pueden ser aprovechadas por otros tipos de datos.

- **Redes Neuronales Profundas**

Una Red Neuronal Profunda es una representación del proceso de aprendizaje de los seres humanos por medio de neuronas en el cerebro, pero en computadores. Una Red Neuronal Profunda sigue la misma idea, sin embargo, generalmente tiene 3 o más capas de neuronas [22]. Las Redes Neuronales Profundas realizan

operaciones más complejas en términos computacionales en comparación a las funciones sigmoides. Se pueden aplicar varios tipos de modelos de aprendizaje profundo en problemas de clasificación de texto.

- **Red Neuronal Convolucional**

En las redes neuronales convolucionales, las circunvoluciones sobre la capa de entrada se utilizan para calcular la salida. Esto da como resultado conexiones locales, donde cada región de la entrada está conectada a una neurona en la salida. Cada capa aplica diferentes filtros y combina sus resultados [23].

2.1.4 Representación de Características para documentos de texto

- **Bolsa de Palabras *(TF - Bag of Words - BoW)***

Este método usado en el procesamiento de lenguaje natural permite representar documentos sin tener en cuenta el orden de las palabras. El objetivo de este modelo es calcular el número de veces que aparece una palabra en un documento [24].

- **Frecuencia de términos - Frecuencia Inversa de los Documentos (TF-IDF Term frequency – Inverse document frequency)**

Este método permite expresar la importancia de una palabra en un documento. Su conclusión se realiza a partir de la cantidad de veces que una palabra se evidencia en un documento y se compara con el número de documentos que mencionan en ese término en el conjunto de datos completo [25].

- ***Hashing***

Este método permite convertir un conjunto de palabras de un documento en un vector por medio de un diccionario, de tal modo que sea fácilmente manipulable. Su disposición puede hacerse por medio de valores numéricos, los cuales asignan un valor a una palabra en particular [26].

- ***Word2vec***

Este modelo usa un sistema de redes neuronales artificiales para asociar las palabras encontradas en un documento con un conjunto de palabras reales. Determina cada palabra diferente en un vector de números [27].

- ***Doc2vec***

Doc2Vec o *Paragraph Vector* es una herramienta de NLP para representación de documentos, la cual genera diferentes vectores para representar la palabra a predecir dentro del documento [28].

Es un marco de trabajo que crea representaciones vectoriales distribuidas continuas para piezas de texto. Los textos pueden ser de longitud variable, desde oraciones hasta documentos. El nombre de *Paragraph Vector* es para enfatizar el hecho de

que el método se puede aplicar a piezas de texto de longitud variable, desde una frase u oración hasta un documento grande [27].

2.1.5 Selección de Características

La selección de características es el proceso de aislar las características más consistentes, no redundantes y relevantes para usar en la construcción de modelos. Reducir metódicamente el tamaño de los conjuntos de datos es importante a medida que el tamaño y la variedad de los conjuntos de datos continúan creciendo. El objetivo principal de la selección de características es mejorar el rendimiento de un modelo predictivo y reducir el costo computacional del modelado.

- **Chi Cuadrado (χ^2)**

En [29] se puede definir a *Chi Cuadrado* como la dependencia entre 2 variables y de la cual puede concluir la relación entre ambas variables. Esta prueba permite concluir si la asociación entre dos variables de una muestra refleja una asociación real con la población.

Considera aspectos como:

- La frecuencia observada (número de observaciones de la clase)
- La frecuencia esperada.

$$x^2{}_c = \sum \frac{(o_i - E_i)^2}{E_i}$$

Donde:

c = grados de libertad.

O = valores observados.

E = valores esperados.

- **Análisis de Componentes Principales (PCA)**

Este método estadístico permite reducir el número de variables en un conjunto de datos teniendo en cuenta la ausencia de correlación entre las variables. Se usa la varianza original para determinar los valores que carecen de una correlación lineal, y su denominación es Componentes Principales [30] [31].

- ***Embeddings Lasso***

En [32] se define el método *Lasso* como una alternativa para realizar la selección idónea de características relevantes para los modelos usados, tomando un subconjunto de las características provistas para la aplicación en el modelo final. El subconjunto definido por *Lasso* es relevante dada la característica de la técnica para forzar los coeficientes de los predictores que tienden a cero. Dado que un predictor con coeficiente de cero no influye en el modelo, Lasso consigue excluir los

predictores menos relevantes. Esto es de suma importancia en la problemática, ya que esto permite que se tomen todas las hojas de vida sin aplicar ningún filtro específico.

2.1.6 Modelos de Ensamble

Los modelos de ensamble utilizan múltiples algoritmos de aprendizaje para obtener un mejor rendimiento predictivo que el que se podría obtener de cualquiera de los algoritmos de aprendizaje por si solos. A diferencia de los modelos clásicos, los modelos de ensamble permiten que exista una estructura mucho más flexible.

A continuación, se definen algunos modelos de ensamble de interés dentro del marco del presente trabajo.

- ***Stacking:***

En [33] se define como una combinación de modelos de clasificación o regresión. Su implementación se resume en la salida de varios modelos, que a su vez constituyen la entrada de otros modelos. Esta propuesta puede generar mejores resultados en términos de rendimiento.

- ***Bagging:***

Según [24], Bagging es una propuesta que gira en torno a la combinación de varios modelos, sin embargo, cada modelo se entrena con subconjuntos del conjunto de entrenamiento. Por último, para dar un resultado general, se realiza un cálculo matemático, el cual suele ser la media aritmética.

- ***Max Voting*:**

En este modelo de ensamble, los submodelos se construyen de forma independiente. Individualmente, se denomina “voto” al resultado de cada submodelo. Como resultado final, Max Voting establece su predicción definitiva gracias a la mayor cantidad de votos recibida [21].

- ***Blending:***

En este modelo en lugar de usar todo el conjunto de datos completo para el entrenamiento, se divide para tener aislados los datos de validación y predicción [34].

- ***Boosting:***

En esta propuesta cada iteración (modelo) permite corregir los errores generados en los modelos anteriores. Esta corrección se establece asignándole un peso mayor a las muestras mal clasificadas y un peso menor a las categorizadas correctamente [35].

2.1.7 Análisis de Texto y Procesamiento de Lenguaje Natural (NLP)

El Análisis de Texto es un proceso mediante el cual se comprende y se extrae información relevante para la toma de decisiones y la combinación con el Procesamiento de Lenguaje Natural por medio de diferentes modelos en herramientas computacionales ha tomado fuerza en diferentes campos, que van desde el análisis de sentimientos, traducciones automáticas, clasificación de contenido hasta la extracción de entidades, tal como se evidencia en [36], [37].

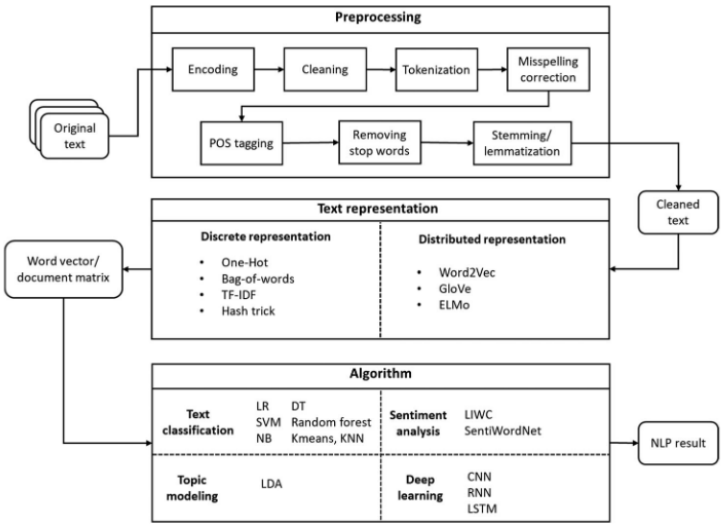
La aplicación de NLP para el procesamiento de textos no estructurados se ha convertido una rama de investigación que ha tomado mucha fuerza en los últimos años, en particular cuando las propuestas de sistemas NLP se soportan con técnicas de Aprendizaje Supervisado, tal como se menciona en [38].

En [39] se realizó una revisión de literatura en el campo de NLP y se concluye que en general, los procesos de NLP se pueden categorizar así:

- Preprocesamiento del texto.
- Representación del texto.
- Modelo de entrenamiento.
- Modelo de evaluación.

En la figura 3, se evidencia el flujo general desde el texto que se debe analizar, hasta los resultados generados por el proceso de NLP.

Figura 3: Flujo de Análisis NLP



Tomado desde: [39].

Tal como se observa en la figura 3, algunas técnicas para el análisis de texto que bien establecidas por la comunidad académica y con buen rendimiento son:

- Regresión Lineal (LR).
- Máquinas de Vectores de Soporte (SVM).
- Bosques Aleatorios (RF).
- Vecinos Cercanos (KNN).
- Árboles de Decisión (DT).
- K-medias (Kmeans).

Además de las técnicas descritas anteriormente, en [40] se resalta con importancia de la aplicación de Redes Neuronales Artificiales en sus diferentes tipos para la aplicación de NLP. En este trabajo, la aplicación de Redes Neuronales Convolucionales arrojó un mayor rendimiento en relación con las demás técnicas analizadas.

2.1.8 Evaluación de Modelos

Para evaluar el rendimiento de modelos de aprendizaje de máquina, se sugiere el uso de métricas de evaluación como*: Precisión (Precision), Exactitud (Accurracy), Exhaustividad (Recall) y Valor F (F1-score)*, que se pueden definir como:

- ***Exactitud*****:** Se define como la cantidad de datos que se clasifican correctamente sobre el total de datos.

$$Exactitud = \frac{TP + TN}{TP + FN + FP + TN}$$

- ***Precisión*****:** Se define como la cantidad de datos devueltos que son correctos.

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

- ***Exhaustividad*****:** se puede definir como la cantidad de datos correctos que se devuelven.

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

- ***Valor F*****:** Es una medida para probar la *precisión*, también conocida como la media harmónica entre la exactitud y la *precisión*.

$$f1 = \frac{2}{\frac{1}{Exhaustividad} + \frac{1}{Precisión}}$$

Dónde

TP = Número de predicciones positivas correctas que son realmente positivas.

FP=Número de predicciones positivas incorrectas que son realmente negativas.

TN= Número de predicciones negativas correctas que son realmente negativas.

FN = Número de predicciones negativas incorrectas que son realmente positivas.

Hay otras métricas de evaluación, tales como *Mean Absolute Error – MAE, Log Loss, Hinge Loss, Quantile Loss, R2, KL Divergence*, pero las aquí mencionadas son de las métricas más comunes para el tipo de modelos empleados en este trabajo.

2.2 Estado del Arte

Para abordar el análisis de texto para currículos, en [41] se propone un algoritmo basado en reglas para la extracción de información relevante de los currículos en China. Los autores analizaron 1500 documentos, los cuales fueron obtenidos desde www.chinahr.com. La clasificación se dividió en 2 grupos, información personal e información profesional. Para el primer grupo, se alcanzó una precisión de 87%, mientras que para el segundo grupo el promedio de precisión fue de 81%. Se resalta la importancia de continuar los estudios para mejorar la precisión obtenida.

En [42] proponen la construcción de un sistema híbrido para el análisis de currículos, el cual se soporta en Máquinas de Vectores de Soporte (SVM) y el Modelo Oculto de Markov (HMM). El estudio realizado por los autores refleja un mejor comportamiento en cuanto a la información personal por parte de SVM, mientras que HMM demostró un mejor rendimiento en términos de información general como los estudios del aspirante.

En [43] proponen el diseño de un algoritmo no supervisado que permite crear automáticamente una gaceta que facilita la creación de etiquetas en diferentes secciones. Como trabajo futuro, se plantea la necesidad de experimentar con algoritmos que permitan implementar gacetas de otros tipos de entidades inmersas en los currículos, como lo son las habilidades, certificaciones, dominios, entre otras.

En [44] proponen una estrategia para clasificar los currículos en 27 categorías diferentes. Se definieron dos modelos diferentes, el primero basado en un método de clasificación “FastText” y segundo en redes neuronales convolucionales (CNN). De allí, la propuesta soportada en CNN mostró mejores resultados en cualquiera de las instancias. Por otra parte, está el uso de NER, en donde recientemente se han explorado métodos basados en el aprendizaje profundo, los cuales han mostrado una mejora significativa en rendimiento y precisión, además de que puede ser usada en un rango más amplio de datos.

En [45] se implementó una CNN para el análisis de 1314 currículos. Los hallazgos de los autores muestran que se pudo elaborar un *dataset* de 5’005.026 descripciones de currículos y puestos de trabajo. El modelo propuesto, alcanzó una precisión del 81.34% y 83.19% bajo las pruebas realizadas.

En [46] aprovechando el uso de redes neuronales artificiales (ANN) para el diseño de una propuesta que identifica los nombres de las instituciones educativas. Su finalidad es la predicción de entidades de educación que no se encuentran debidamente etiquetadas. La precisión en esta propuesta fue del 92.06%. Como trabajo futuro, se plantea la aplicación del algoritmo a otros componentes de la sección educativa en un currículo.

En [47] se presenta un modelo que permite detectar las habilidades profesionales expuestas en los currículos. Por medio de ontologías, el algoritmo detecta si existen habilidades asociadas, o si, por el contrario, no se encuentran presentes. Los autores definen un total de 13.337 habilidades definidas en la ontología inicial, y plantean la necesidad de extender sus investigaciones a documentos en otros idiomas.

En [48] se desarrolló un sistema soportado en Redes Neuronales Artificiales donde realiza la clasificación de entidades nombradas. Los autores resaltan la relevancia del sistema para usuarios con poca experiencia en el campo del NER.

2.3 Discusión sobre Revisión de Literatura

Los procesos exploratorios han permitido observar que el análisis de currículos puede ser logrado con un porcentaje de precisión elevado haciendo uso de técnicas de Inteligencia Artificial. Por otro lado. Algunos autores plantean la importancia de implementar otras estrategias soportadas en diferentes técnicas de Aprendizaje Supervisado, Aprendizaje No Supervisado y Aprendizaje Semi - Supervisado, lo que permite concluir, que a pesar de los resultados con porcentajes de precisión altos, la mayoría de propuestas únicamente han sido probadas en segmentos o secciones específicas de un currículo, lo que implica, un aporte valioso para la comunidad, la realización de más pruebas con diferentes enfoques.

En el problema de minería de datos, el rendimiento del modelo depende de los datos. Naïve Bayes es un modelo probabilístico y el modelo más utilizado para la clasificación de texto y proporciona una precisión mejorada en conjuntos de datos pequeños e independientes. Si los datos están correlacionados, entonces se puede usar SVM. Linear SVC es similar a SVM con parámetro de kernel “lineal” e implementado en términos de liblinear en lugar de libsvm, por lo que tiene más flexibilidad en la elección de penalizaciones y funciones de pérdida. GridSearchCV toma un diccionario que describe los parámetros que se podrían probar en un modelo para entrenarlo. En nuestro experimento, hemos combinado Grid Search con SVM para definir hiperparámetros y probado múltiples combinaciones de hiperparámetros. Una Red Neuronal Artificial es un perceptrón simple que funciona bien para un conjunto de datos simple, se pueden introducir más capas ocultas para mejorar el rendimiento. El bosque aleatorio es un algoritmo de aprendizaje supervisado y no se ajusta a la clasificación de texto solo con la información del CV del solicitante. La regresión logística utiliza un solucionador de descenso de gradiente, mientras que en SGD Classifier, utiliza el descenso de gradiente estocástico que converge más rápido que el descenso de gradiente. Elastic Net ha mostrado un rendimiento significativamente mejorado en comparación con otros modelos de ML. Lasso (L_1) y Ridge(L_2) se utilizan como regularizador en la Regresión Lineal llamada Elastic Net. La regularización ayuda a minimizar la función de pérdida ajustada y evita el sobreajuste o sub-ajuste en el entrenamiento del modelo.

Los modelos de aprendizaje profundo han requerido un gran conjunto de datos para el entrenamiento, donde el conjunto de datos en español consta de 642 muestras. Estos son los datos con los que se contó para adelantar el proceso de investigación. Hemos utilizado LabelEncoding para convertir etiquetas de destino de cadena a numéricas para entrenar modelos de aprendizaje profundo. Se realizaron los siguientes experimentos utilizando Conteo de Vectores, Keras-Tokenization-Freq (Word-Embedding) y word2vec previamente entrenados como características y entrenado la red utilizando Dense Neural Network y CNN.

3 Metodología para Clasificación de Currículos con NLP (CVNLP)

3.1 Introducción

De las diferentes técnicas de aprendizaje automático se ha podido observar que los mecanismos de clasificación supervisada es la mejor alternativa para la presente investigación, teniendo en cuenta que el objetivo es asistir al personal encargado de la revisión de hojas de vida a encontrar al candidato idóneo. Sin embargo, a pesar de que existen varias metodologías generales de analítica de datos (ejemplo: CRISP-DM), se considera adecuado realizar una adaptación de dichas metodologías en conjunto con los métodos apropiados.

El alcance de esta adaptación para el análisis de currículos es detallar precisamente las diferentes técnicas de preparación de texto, selección de modelos supervisados clásicos y modelos supervisados avanzados basados en redes neuronales, para que una herramienta automática o una persona pueda utilizar los resultados anteriores para facilitar y apoyar el análisis y selección de currículos de acuerdo con unas necesidades específicas.

Teniendo en cuenta la metodología CRISP-DM como referencia, se contemplan las siguientes etapas:

1. Caracterización del conjunto de datos
2. Análisis exploratorio y preparación de los datos.
3. Representación y Selección de características
4. Entrenamiento de los modelos
5. Refinamiento
6. Evaluación de la metodología
7. Despliegue.

A continuación, se describe detalladamente el alcance de cada etapa y el conjunto de técnicas y métodos implementados.

3.1.1 Caracterización del conjunto de datos

Las hojas de vida contemplan un amplio rango de información que cubre desde la información personal, hasta la formación académica y profesional. El formato bajo el cual se presentan puede variar, y por esta razón se puede dificultar su lectura y posterior análisis a nivel computacional.

En una hoja de vida es común encontrar:

- Información personal
- Información sociodemográfica
- Información de académica
- Habilidades generales y especificas
- Experiencia laboral

No obstante, debido al alcance del proyecto y el enfoque orientado en un 100% a datos no estructurados, se usará el contenido de la hoja en su totalidad. Buscando mejorar la precisión del modelo, se generó una segmentación por cada una de las categorías.

Los documentos de texto de hojas de vida se pueden dividir en las siguientes 3 categorías:

1. **No estructurados:** cada hoja de vida es un documento que presenta toda la información de un documento en formato libre, sin ninguna regularidad ni esquema que permita identificar una sección específica.
2. **Semiestructurados**: son formatos de archivos donde se puede identificar sintácticamente cada una de las secciones de una hoja de vida, pero al interior de cada sección es texto puro sin ninguna estructura regular.
3. **Estructurada:** son archivos texto en los cuales es posible identificar sintácticamente cada una de las secciones de una hoja de vida y dentro de cada sección, se puede identificar completamente cada ítem y tipo de entrada, suelen requerir una estructuración definida y tienen una etiqueta.

Desde el punto de vista del análisis y facilidad de procesamiento de texto, los archivos de preferencia son en orden: 3, 2, y 1.

La precisión de los resultados depende en gran medida del tipo de formato de archivo descrito anteriormente, los mejores resultados se deben obtener con los formatos 3, luego 2 y por último 1.

Para el efecto de aplicación de esta metodología, se identificarán los datos con el tipo 1 (texto no estructurado), 2 (texto semi-estructurado), 3 (texto estructurado).

Dado el alcance definido en este proyecto, sólo serán considerados los datos en formato no estructurados.

Un previo análisis exploratorio sobre los archivos permite entender mejor la naturaleza de los datos sobre los cuales se está trabajando, adicionalmente, facilita la transformación los datos para ser usados a futuro. De allí la importancia de realizar un conteo de registros, validar registros que sean nulos, como abordarlos y posteriormente, cómo manipularlos.

3.1.2 Análisis exploratorio y preparación de los datos:

3.1.2.1 Análisis exploratorio de los datos

El análisis exploratorio de los datos se define como una etapa que permite a la persona validar la calidad de los datos y la relevancia de estos para el proyecto que se desea efectuar, además permite encontrar muchos *insights*, información o conocimiento en los mismos datos.

Para el caso de la calidad de los datos, es necesario realizar un proceso exploratorio de los datos con los siguientes propósitos:

a) Eliminar datos irrelevantes que puedan generar una carga computacional adicional durante el procesamiento de los datos

b) Refinar el conjunto de datos que se usará para trabajar
c) Depurar la cantidad de información que será usada.
d) Validar los inconvenientes que puedan surgir debido a: la cantidad de los datos, la completitud o la naturaleza.

3.1.2.2 Eliminación de datos irrelevantes

En primera instancia es necesario identificar qué tipo de análisis se realizará sobre el conjunto de datos, y posteriormente eliminar una columna irrelevante del tabular que no contribuiría o sería útil para el entrenamiento del modelo ML/DL.

3.1.2.3 Tokenización

La tokenización corresponde al proceso donde se separan las palabras del texto en componentes más simples llamados *tokens.* Los tokens son los componentes básicos del lenguaje natural. El objetivo principal de la tokenización es crear un vocabulario o corpus.

Ejemplo = “Este es un uso de tokenizacion usado en este ejemplo”

Resultado = [‘Este’, ‘es’, ‘un’, ‘uso’, ‘de’, ‘tokenizacion’, ‘usado’, ‘en’, ‘este’, ‘ejemplo’]

3.1.2.4 Manejo de datos faltantes

En múltiples ocasiones, es necesario validar la calidad de los datos, es por esto que es necesario entrar a entender si la completitud que presenta los datos es suficiente o es necesario trabajar sobre este conjunto de datos.

Es decir, en la mayor parte de conjuntos de datos usados para texto, entender si un registro está incompleto puede significar una mejoría considerable.

En este proyecto, entender si un currículo contiene el perfil profesional o no, definiría si el registro es válido.

3.1.2.5 Normalización de texto

Es el proceso de estandarizar el texto, por lo que el modelo ML / DL comprende mejor la entrada humana. Este proceso también se denomina normalización de textos.

Algunos de las técnicas más usadas para esto son convertir todo en mayúsculas o minúsculas.

Ejemplo = [‘Este’, ‘es’, ‘un’, ‘uso’, ‘de’, ‘tokenizacion’, ‘usado’, ‘en’, ‘este’, ‘ejemplo’]

Resultado = [‘este’, ‘es’, ‘un’, ‘uso’, ‘de’, ‘tokenizacion’, ‘usado’, ‘en’, ‘este’, ‘ejemplo’]

Como se aprecia en el resultado, el token 1 y el 9 ahora son iguales, lo que facilita el tratamiento de los datos.

3.1.2.6 Quitar palabras de parada, espacios en blanco y signos puntuación

Para limpiar los datos, es importante remover palabras de parada y puntuación que no agregan valor y pueden arrojar mejores métricas en fase de entrenamiento de modelos ML/DL.

Ejemplo = [‘este’, ‘es’, ‘un’, ‘uso’, ‘de’, ‘tokenizacion’, ‘usado’, ‘en’, ‘este’, ‘ejemplo’]

Resultado = [‘uso’, ‘tokenizacion’, ‘usado’, ‘ejemplo’]

Aquí se transforma una matriz de 10 tokens en 4 tokens que poseen mayor relevancia.

3.1.2.7 Eliminar caracteres Unicode

Los emojis, las URL y las @ es ruido para los modelos de ML/DL porque son firmas únicas que terminan traduciéndose inútilmente a Unicode y actúan como ruido en los datos. Este paso es aplicado en el análisis sintáctico de hojas de vida debido a la naturaleza profesional de los textos, no obstante, hay casos particulares, como el análisis de texto en redes sociales.

3.1.2.8 Etiquetado de partes del habla (POS)

El etiquetado POS permite asociar una etiqueta que representa un elemento sintáctico en un lenguaje (como verbos, sujetos, sustantivos, adjetivos, adverbios, etc). Esto permite calcular los descriptores más y menos utilizados y realizar análisis utilizando datos etiquetados. Cada parte del discurso tiene su propia etiqueta POS única. El uso del etiquetado POS se puede emplear para continuar con el filtrado de datos, eliminando verbos sin relevancia, adjetivos o sustantivos con poca frecuencia, etc.

3.1.2.9 Derivación (Stemming) - Lematización (Lemmatization)

La Derivación es el proceso por el cual se puede generar una reducción de una palabra a su raíz. La *lematización* se centra en disminuir, aún más, el conjunto de datos. Por esta razón, es importante la aplicación de estas técnicas, para transformar palabras a su raíz más básica. Esto incluye conjugaciones de verbos, plurales, entre otros.

Ejemplo = [’correr’, ‘corre’]
Resultado = [’corr’, ‘corr’]

3.1.3 Representación y Selección de características

En este paso, los datos de texto limpios después del proceso de *Lematización*, se transforman en vectores de características. A continuación, se presentan diferentes técnicas de representación de documentos para NLP.

3.1.3.1 Vectores de Recuento (Count Vectors)

Los vectores de recuento representan una notación matricial del conjunto de datos en la que cada fila representa una muestra de datos del texto limpiado y cada

columna representa un término de los datos, y cada celda representa el recuento de frecuencia de un término en particular.

3.1.3.2 Vectores TF-IDF (TF-IDF Vectors)

La puntuación TF-IDF representa la importancia relativa de un término en el conjunto de datos. La puntuación TF-IDF está compuesta por dos términos: el primero calcula la Frecuencia de Término Normalizada (TF), el segundo término es la Frecuencia inversa del Documento (IDF), calculada como el logaritmo del número de documentos en el corpus dividido por el número de documentos donde aparece el término específico.

$$TF(t) = \frac{(Number\ of\ times\ term\ t\ appears\ in\ a\ dataset)}{Total\ number\ of\ terms\ in\ dataset}$$

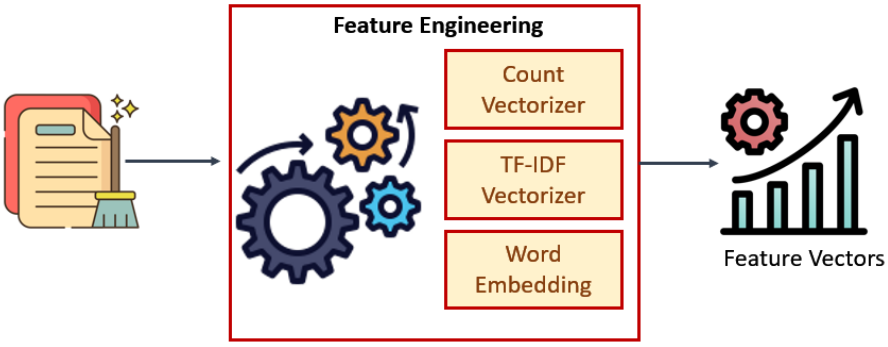
$$IDF(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it})$$

Los vectores TF-IDF se pueden generar en diferentes niveles de tokens de entrada.

3.1.3.3 Incrustación de Palabras (Word Embeddings)

Una incrustación de palabras es una forma de representar palabras utilizando una representación vectorial densa. La posición de una palabra dentro del espacio vectorial se aprende del texto y se basa en las palabras que rodean a la palabra cuando se usa. Las incrustaciones de palabras se pueden entrenar utilizando el propio corpus de entrada o se pueden generar mediante incrustaciones de palabras previamente entrenadas, como Word2Vec.

Figura 4: Ingeniería de características de los datos de texto



Fuente: Elaboración propia

3.1.4 Entrenamiento de los modelos de clasificación

La clasificación de texto consiste en entrenar a un algoritmo utilizando las características creadas en el paso anterior. Se pueden utilizar múltiples modelos

de aprendizaje automático para el entrenamiento. Algunos de los clasificadores que pueden ser usados para este propósito son:

1 Clasificador Naïve Bayes.
2 Máquinas de vectores de soporte.
3 Regresión logística.
4 Bosque Aleatorio.
5 Descenso Gradiente Estocástico

3.1.5 Refinamiento

En este punto se validan los resultados de los modelos entrenados previamente, para determinar si son adecuados a la necesidad existente. En caso de que los modelos no sean óptimos, se debe realizar un ajuste de los procesos mencionados anteriormente y determinar si el uso de otras técnicas puede facilitar la implementación de los modelos.

Algunas de las técnicas que pueden ser usadas para refinar los modelos son:

a. Ajuste fino (*fine tuning*): es un proceso en el que se refinan los parámetros de entrada de los modelos, es decir, se modifican los parámetros de los diferentes modelos de entramiento
b. Se evalúan diferentes modelos de selección de características o se evalúa la calidad de la limpieza de los datos.
c. Se validan diferentes modelos, algunas opciones son modelos de ensamble, en los cuales se combinan diferentes modelos pre entrenados en uno solo, lo cual convierte modelos endebles en un solo modelo más robusto.

3.1.6 Evaluación de la metodología

Para validar la efectividad de la metodología, es necesario comprobar el funcionamiento con un conjunto de datos diferente. Por esta razón, se sugiere replicar el proceso con un subconjunto de datos o con un conjunto de datos similar.

3.1.7 Despliegue de modelos.

El despliegue o *deployment* de modelos de aprendizaje automático, son un conjunto de actividades que buscan que un sistema esté disponible para los usuarios, en este caso, que otros sistemas puedan tener acceso a diferentes modelos, para que puedan recibir datos y devolver sus predicciones.

La implementación del modelo se determina desde a definición del proyecto, donde se sugiere el uso de una metodología.

4 Aplicación de la Metodología CVNLP

La aplicación busca sustentar por medio de pruebas y evidencias que los pasos definidos en la metodología son coherentes y dan solución al problema de investigación definido en el presente proyecto. A continuación, se describen los pasos realizados.

4.1.1 Descripción de los datos

El conjunto de datos que se usará para la primera validación de la metodología consiste en un conjunto de 725 hojas de vida previamente agrupadas en carpetas las cuales se encuentran divididas según las características del perfil profesional de cada persona.

Estas hojas de vida se encuentran en los siguientes formatos:

- PDF
- DOCX
- DOC

Adicionalmente, ninguna de estas hojas de vida se filtró, ya que el objetivo de la metodología es la aplicación de manera transversal a cualquier conjunto de datos (siempre y cuando se encuentre relacionado al objetivo del trabajo), sin necesidad de aplicar ninguna limpieza o filtro adicional.

En estas hojas de vida se pueden encontrar las siguientes secciones o apartados, no obstante, no tienen ninguna etiqueta (son datos 100% no estructurados):

- Información personal
- Información sociodemográfica
- Información de académica
- Habilidades generales y especificas
- Experiencia laboral.

El proceso inicial requiere que este conjunto de datos se transforme a un formato que pueda ser usado para el análisis y transformaciones correspondientes, de allí, surge la necesidad de transformar todo el texto dentro de cada documento y el nombre de la carpeta donde se encuentra. De esta manera se crea un registro de 1 fila y 2 columnas, que luego se anexa secuencialmente el nombre de la carpeta que se usó con etiqueta o *label*.

Este proceso es necesario únicamente en el caso de que los datos que se tengan no estén previamente estructurados o que el formato en el que se encuentren no sea de uso para el proyecto.

4.1.2 Análisis exploratorio y preparación del conjunto de datos:

4.1.2.1 Análisis exploratorio de los datos

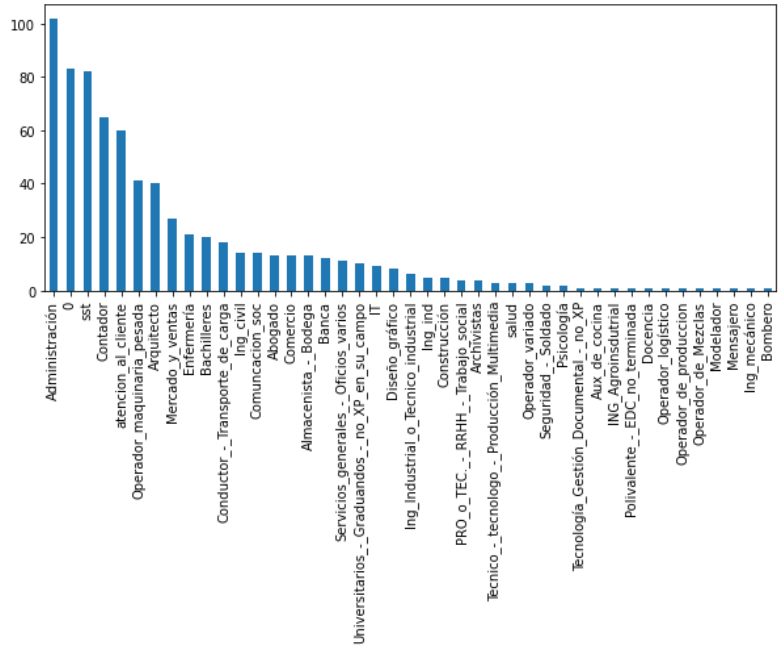
Durante el proceso de análisis exploratorio de los datos se encuentra un conjunto de datos con las siguientes características:

El conjunto de datos está compuesto por una matriz de 725 registros con dos columnas cada uno (Texto, Clase). En el 'Texto' se encontró cada uno de los archivos y el 'Clase' el nombre de directorio que actuará como *label* o etiqueta para el modelo de entrenamiento posterior.

Dentro de estos datos, se encuentra que:

1. 83 registros tienen un valor de 0, lo cual convierte a estas instancias en valores inválidos.
2. Hay clases que tienen muy poca representación, lo que genera un desbalanceo de clases.

Figura 5: Frecuencia de clases



Fuente: Elaboración propia

4.1.2.2 Eliminar datos irrelevantes

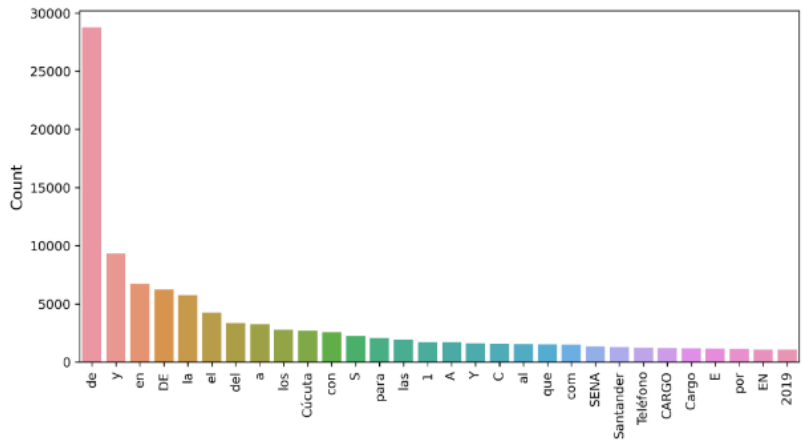
En esta primera fase se eliminan registros que no aportan ningún valor y que se logra identificar a mediante el análisis exploratorio, el resultado es una matriz de 642 registros y dos columnas.

4.1.2.3 Tokenización

En este proceso de tokenización, se puede hacer uso de diferentes herramientas, no obstante, se opta por el uso de nltk como librería principal para el tratamiento de la información.

Este proceso de tokenización permite transformar un registro inicial, a un registro que se puede utilizar en un modelo de procesamiento de lenguaje natural, convirtiendo la información que se encuentra dentro cada texto en unidades de procesamiento semántico.

Figura 6: Conteo de tokens preliminar.



Fuente: Elaboración propia

4.1.2.4 Normalizar texto

La normalización del texto puede tomarse como un paso de entrada, que surge de la necesidad de generar un corpus homogéneo con el mismo conjunto de características.

Entre las técnicas aplicadas se encuentra volver minúsculas las palabras y eliminar acentos.

No se enfoca en disminuir el tamaño de la matriz de características.

Antes: 36386 tokens.

Después: 36386 tokens.

4.1.2.5 Quitar palabras de parada, espacios en blanco y signos puntuación

Con el conjunto de datos normalizado, se procede a eliminar palabras de parada, signos de puntuación, espacios en blanco y cualquier otro *token* que no se considere relevante, por ejemplo: *'me','esa','el','la','los'*

También se pueden eliminar palabras específicas del conjunto de datos, entre las cuales se encuentran:

- Cúcuta
- Norte de Santander
- Teléfono.

Adicionalmente, se eliminan signos de puntuación que puedan llegar a ser considerados como *token,* estos no tienen relevancia para los modelos de entrenamiento y puede afectar los resultados y métricas de precisión.

Antes: 36386 tokens.

Después: 14383 tokens.

4.1.2.6 Eliminar caracteres Unicode

Ahora bien, los caracteres mencionados anteriormente no son los únicos que pueden representar un inconveniente en el entrenamiento de los modelos, actualmente se encuentran algunos caracteres especiales (👆) que pueden generar incongruencias o dificultades no previstas originalmente en el entrenamiento de los modelos, es por esto por lo que se procede a eliminar todo carácter Unicode del texto.

Antes: 14383 tokens.

Después: 14364 tokens.

4.1.2.7 Etiquetado de partes del habla (POS)

El uso del etiquetado POS se puede definir con fines exploratorios, para validar relevancia, repetición o uso de ciertas palabras, no obstante, no fue necesario su implementación en esta investigación, puesto que la relevancia del experimento se centraba en el uso del conjunto de datos como un total y no de la exploración individual de cada *token.*

4.1.2.8 Stemming y Lemmatization

En este último paso se transforman las palabras a una forma más básica, haciendo que un conjunto de palabras conformado por diferentes verbos y sus distintas conjugaciones, se conviertan en uno solo.

Una vez finalizado el proceso de preparación de datos, encontramos un corpus más coherente, que tiene relación al conjunto de datos y sirve de entrada para el entrenamiento de los modelos.

Antes: 14364 tokens.

Después: 14364 tokens.

Dado que los procesos de Stemming y Lemmatization modifican la estructura de la palabra pero no la cantidad, el número de tokens sigue siendo el mismo.

4.1.3 Representación y Selección de características

En este paso, los datos de texto limpios (después de la lematizar) se transformarán en vectores de características. A continuación, se describe detalladamente los pasos a seguir para obtener características relevantes del conjunto de datos.

Como referencia, se hace uso de la biblioteca de aprendizaje automático scikit-learn en Python.

De aquí en adelante todos los pasos se harán de dos maneras:

Método A: Se tomarán los datos completos, sin filtrar o seleccionar características.

- Este método contiene 642 registros y 42 clases

Método B: Se hará un tratamiento adicional que consiste en:

- Se define un umbral de registros mínimos por clase.
- Se limitarán la cantidad de características de los modelos a 1000 para hacer más robusto el entrenamiento. (Matriz de 417 por 1000 características)
- Este método contiene 417 registros y 7 clases (como es el balanceo de clases)

4.1.3.1 Count Vectors (Conteo de vectores)

Método A:

En este primer modelo de representación de características se hace uso de *Conteo de Vectores,* aquí tenemos como resultado inicial, una matriz de 642 registros por 36386 características.

Método B:

En este segundo modelo de representación de características se hace uso de *Conteo de Vectores*, aquí tenemos como resultado inicial, una matriz de 417 registros por 1000 características.

4.1.3.2 TF-IDF Vectors

Método A:

En este modelo de representación de características se hace uso de TD-IDF, aquí tenemos como resultado, nuevamente se obtiene una matriz 642 registros por 36386 características.

Método B:

Por otro lado, haciendo uso de este mismo método, se obtiene una matriz de 417 registros por 1000 características, previamente definidas.

4.1.4 Entrenamiento de los modelos

La idea es utilizar múltiples modelos de aprendizaje automático para el entrenamiento y mediante técnicas de evaluación de modelos, seleccionar el más adecuado sin olvidar la naturaleza del conjunto de datos. Ahora bien, los modelos aquí mostrados no son más que una muestra del trabajo total, aquí se resumen algunos de los modelos que se pueden usar.

Es importante destacar que los resultados mostrados a continuación corresponden exclusivamente a los resultados de los modelos usando los vectores generados por *Count Vectorizer*.

Tabla 1: Evaluación de rendimiento del conjunto de datos inicial con modelos de Aprendizaje de Máquina haciendo uso de la Metodología A

Model	Accuracy (%)	Precision (%)	Recall (%)
Naïve Bayes	30	6	16
SVM	34	13	34
SGD	55	34	47
Grid Search with SVM	51	39	51
Linear SVC	**53**	**34**	**53**
Random Forest	49	28	49
Logistic Regression	35	10	26

Fuente: Elaboración propia

La tabla 1 muestra que SGD y SVC poseen los mejores resultados, no obstante, es bastante alejado del resultado esperado de un modelo, por lo cual es necesario validar alternativas para mejorar el rendimiento.

Tabla 2: Evaluación de rendimiento del conjunto de datos inicial con modelos de *Aprendizaje de Máquina* haciendo uso de la Metodología B.

Model	Accuracy (%)	Precision (%)	Recall (%)
Naïve Bayes	53	48	50
SVM	48	56	48
SGD	68	63	64
Grid Search with SVM	63	62	63
Linear SVC	**70**	65	**66**
Random Forest	48	56	48
Logistic Regression	60	56	60

Fuente: Elaboración propia

La Tabla 2 muestra que, entre los modelos de aprendizaje automático utilizados, Linear SVC ha tenido el mejor desempeño. En los problemas de análisis de texto, el rendimiento del modelo depende de los datos. Si los datos están correlacionados, entonces se puede usar SVM. SVC lineal es similar a SVM con parámetro de kernel "lineal" e implementado en términos de liblinear en lugar de libsvm, por lo que tiene más flexibilidad en la elección de penalizaciones y funciones de pérdida. GridSearchCV toma un diccionario que describe los hiperparámetros que se podrían probar en un modelo para entrenarlo. En nuestro experimento, hemos combinado Grid Search con SVM para definir hiperparámetros y probado múltiples combinaciones de hiperparámetros.

Los modelos de aprendizaje profundo requieren de un gran conjunto de datos para el entrenamiento, desafortunadamente el conjunto de datos consiste únicamente con 642 muestras. Sin embargo, hay redes como las CNN y los mecanismos de *transfer learning* que permiten para aplicaciones principalmente de imágenes (aunque en texto también) de utilizar redes pre entrenadas o terminar de entrenar su última capa. O modelos pre entrenados como BERT

La Tabla 2 muestra el rendimiento de los modelos de *Deep Learning* entrenados.

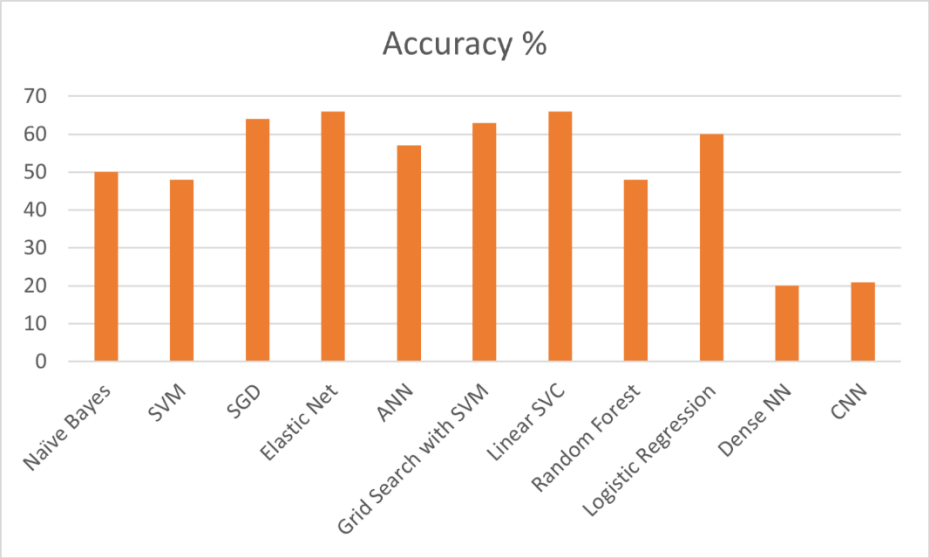
Tabla 3: Evaluación de rendimiento del conjunto de datos inicial usando modelos de *Deep learning*.

Feature Engineering	Model	Accuracy (%)
Count Vectors	Dense NN	20%
Tokenization-Word Embedding	Dense NN	20%
Spanish Word2Vec	Dense NN	20%
Tokenization-Word Embedding	CNN	21%

Fuente: Elaboración propia

La tabla 3 muestra que los resultados obtenidos no son relevantes, puesto que el tamaño del conjunto de datos es insuficiente para hacer uso de modelos de *Deep Learning*.

Figura 7: Comparación del rendimiento de las modelos



Fuente: Elaboración propia

4.1.5 Refinamiento

Para visualizar la relevancia del refinamiento de los modelos, es necesario devolverse a los pasos anteriores, donde el **método B** es el que tiene medidas de refinamiento, lo cual permite tener resultados robustos.

Se hizo de esta manera para poder evaluar y comparar el impacto que puede tener en un proyecto de NLP refinar y hacer retroalimentación sobre los propios pasos para generar mejores resultados.

4.1.6 Evaluación de la metodología

Una vez desarrollada la metodología, se procede a comparar los resultados conseguidos, haciendo uso del **Método B** contra los resultados hallados en un conjunto de datos relacionado.

Se repitió el experimento haciendo uso de un conjunto de datos de Kaggle que consiste en 122519 registros y los resultados se encuentran en la tabla 4.

Tabla 4: Evaluación de rendimiento en un conjunto de datos diferentes con la metodología diseñada.

Model	Accuracy (%)	Precision (%)	Recall (%)
Naïve Bayes	91	91	91
SVM	94	94	94
SGD	87	88	87

ANN	91	91	91
Grid Search with SVM	87	88	87
Linear SVC	93	93	93
Random Forest	91	91	91
Logistic Regression	93	93	93

Fuente: Elaboración propia

4.1.7 Despliegue de los Modelos.

En el caso puntual de esta implementación de la metodología, el despliegue se hará en un ambiente local controlado por parte de la empresa que proporcionó los datos originales.

El uso principal de la metodología se enfocará en:

1. Ayudar en la gestión de nuevas hojas de vida.
2. Servir como doble validación a la hora de realizar la clasificación manual.
3. Generar un primer filtro en la selección de perfiles.

No obstante, el modelo, en este caso puntual, será reentrenado en varias ocasiones, con el objetivo de permitir un mayor rango de perfiles profesionales.

5 Conclusiones y Trabajo Futuro

5.1 Conclusiones.

Del análisis realizado en el marco de la investigación se ha concluido que el seguimiento de una metodología enfocada en un área de proyectos puede generar mejores resultados, puesto que se propone un conjunto de pasos, o metodología, que facilita conseguir resultados a partir de la implementación de esta.

Adicionalmente, se consigue que la metodología consiga abordar un múltiple conjunto de datos, con resultados favorables para la investigación.

Esto incluye la posibilidad de usar un conjunto de datos con cualquier naturaleza, puesto que como objetivo principal tiene que sea aplicable de manera transversal a diferentes escenarios y diferentes condiciones; ya sea conjuntos de datos pequeño y simple, hasta un conjunto de datos especializado.

Se evidencia que la correcta clasificación de los registros, así como el acertado manejo de clases subrepresentadas es clave para llegar a obtener resultados deseables.

5.2 Trabajo futuro

Se sugiere el uso de NER (Named Entity Recognition) en combinación con los modelos de clasificación, para extraer un conjunto de características más relevantes para el conjunto de datos.

Se plantea el uso del etiquetado POS para extraer características relevantes para cada clase dentro del conjunto de datos y combinarlo con un sistema de extracción de texto.

La creación de un conjunto de datos sintético, a partir de los datos existentes, en los casos donde no se cuente con una cantidad suficiente de información podría plantearse como una posible solución a la hora de implementar modelos de Aprendizaje profundo

6 Referencias

[1] D. Torres-Flórez, J. S. Velasquez-Díaz, and J. W. Hernández-Gónzalez, “Importancia del reclutamiento y la selección del personal en el sector hotelero: Caso Villavicencio-Colombia,” *Desarrollo Gerencial*, vol. 12, no. 1, pp. 1–23, Jun. 2020, doi: 10.17081/dege.12.1.3619.

[2] A. B. Calibar, J. Holleger, and R. O. Klenzi, “Análisis de Similitud en Documentos de Texto Mediante Técnicas de Ciencia de Datos Basadas en Aprendizaje Profundo (Deep Learning),” pp. 246–250, 2018.

[3] C. Mendez, C. C. Ordoñez Quintero, H. Ordoñez, and A. Ordoñez, “Sistema de Indexación de Documentos Jurisprudenciales Soportado enInteligencia Artificial,” pp. 41–52, 2019.

[4] E. Barrientos-Avendaño, A. Coronel, U. Francisco, P. Santander, and D. Rico-Bautista, “Fabián Cuesta Quintero,” 2020. [Online]. Available: https://www.researchgate.net/publication/339227416

[5] H. Aguilar and T. M. Baca, “Reclutamiento y selección de personal para fortalecer la gestión estratégica para la empresa BlueCall,” *UNIVERSIDAD NACIONAL DE TRUJILLO*, 2021.

[6] G. Galeano and C. Gámez, “La inteligencia artificial en los procesos de selección,” *Universidad de Valladolind*, 2021.

[7] J. Fernandez, N. Miranda, R. Guerrero, and F. Piccoli, “Datos no Estructurados No Textuales: Desarrollo de Nuevas Tecnologías,” *WICC 2010 - XII Workshop de Investigadores en Ciencias de la Computación*, pp. 330–336, 2010.

[8] B. Balducci and D. Marinova, “Unstructured data in marketing,” *Journal of the Academy of Marketing Science*, vol. 46, no. 4. Springer New York LLC, pp. 557–590, Jul. 01, 2018. doi: 10.1007/s11747-018-0581-x.

[9] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A Survey on Distributed Machine Learning,” *ACM Computing Surveys*, vol. 53, no. 2. Association for Computing Machinery, Jun. 01, 2020. doi: 10.1145/3377454.

[10] R. Sathya and A. Annamma, “THE SCIENCE AND INFORMATION ORGANIZATION INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE,” vol. 2, pp. 34–38, 2013, [Online]. Available: www.ijarai.thesai.org

[11] P. Jadon, D. Bhatia, and D. Kumar Mishra, *A BigData approach for sentiment analysis of twitter data using Naive Bayes and SVM Algorithm*. 2019.

[12] S. Xiao and W. Tong, “Prediction of User Consumption Behavior Data Based on the Combined Model of TF-IDF and Logistic Regression,” in *Journal of Physics: Conference Series*, Feb. 2021, vol. 1757, no. 1. doi: 10.1088/1742-6596/1757/1/012089.

[13] M. Choubisa, R. Doshi, N. Khatri, and K. K. Hiran, “A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security,” in *2022 International Conference on IoT and Blockchain Technology, ICIBT 2022*, 2022. doi: 10.1109/ICIBT52874.2022.9807766.

[14] M. M, R. A, and E. bakry HM, “An Efficient Classification Model for Unstructured Text Document,” *American Journal of Computer Science and Information Technology*, vol. 06, no. 01, 2018, doi: 10.21767/2349-3917.100016.

[15] M. R. Davahli *et al.*, “Identification and prediction of human behavior through mining of unstructured textual data,” *Symmetry*, vol. 12, no. 11. MDPI AG, pp. 1–23, Nov. 01, 2020. doi: 10.3390/sym12111902.

[16] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Mach Learn*, vol. 109, no. 2, pp. 373–440, Feb. 2020, doi: 10.1007/s10994-019-05855-6.

[17] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, “A Simple Semi-Supervised Learning Framework for Object Detection,” May 2020, [Online]. Available: http://arxiv.org/abs/2005.04757

[18] M. Abukmeil, S. Ferrari, A. Genovese, V. Piuri, and F. Scotti, “A Survey of Unsupervised Generative Models for Exploratory Data Analysis and Representation Learning,” *ACM Computing Surveys*, vol. 54, no. 5. Association for Computing Machinery, Jun. 01, 2021. doi: 10.1145/3450963.

[19] R. Salakhutdinov, “Learning Deep Generative Models,” *Annu Rev Stat Appl*, vol. 2, pp. 361–385, Apr. 2015, doi: 10.1146/annurev-statistics-010814-020120.

[20] M. A. Espejel-Rivera, R. Calderón-Suárez, R. M. Ortega-Mendoza, C. J. Camacho-Bello, and M. A. Máquez-Vera, “Detección automática de noticias falsas usando representaciones textuales tradicionales y soluciones basadas en aprendizaje profundo,” *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 10, no. Especial3, pp. 120–127, Aug. 2022, doi: 10.29057/icbi.v10iespecial3.9008.

[21] S. Yeon Yoo, H. Ju Lee, and S. Oh, “Role of unstructured data on water surface elevation prediction with LSTM: case study on Jamsu Bridge, Korea,” *J. Korea Water Resour. Assoc*, vol. 54, no. 1, pp. 1195–1204, 2021, doi: 10.3741/JKWRA.2021.54.S-1.1195.

[22] P. Kopper, S. Pölsterl, C. Wachinger, B. Bischl, A. Bender, and D. Rügamer, “Semi-Structured Deep Piecewise Exponential Models,” SP-ACA, 2021.

[23] M. Carbonell, P. Riba, M. Villegas, A. Fornés, and J. Lladós, “Named entity recognition and relation extraction with graph neural networks in semi structured documents,” in *Proceedings - International Conference on Pattern Recognition*, 2020, pp. 9622–9627. doi: 10.1109/ICPR48806.2021.9412669.

[24] H. Iwasaki, Y. Chen, A. H. Huang, and H. Wang, “Neural Network Translated into Bag-of-Words: Lexicon of Attentions,” 2018.

[25] M. Das, S. Kamalanathan, and P. J. A. Alphonse, “A Comparative Study on TF-IDF Feature Weighting Method and its Analysis using Unstructured Dataset,” 2020.

[26] J. Xie, C. Qian, D. Guo, M. Wang, G. Wang, and H. Chen, “COIN: An Efficient Indexing Mechanism for Unstructured Data Sharing Systems,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 313–326, Sep. 2021, doi: 10.1109/tnet.2021.3110782.

[27] Z. Xu *et al.*, “A text-driven aircraft fault diagnosis model based on a word2vec and priori-knowledge convolutional neural network,” *Aerospace*, vol. 8, no. 4, Apr. 2021, doi: 10.3390/aerospace8040112.

[28] H. B. Dogru, S. Tilki, A. Jamil, and A. Ali Hameed, “Deep Learning-Based Classification of News Texts Using Doc2Vec Model,” in *2021 1st International Conference on Artificial Intelligence and Data Analytics, CAIDA 2021*, Apr. 2021, pp. 91–96. doi: 10.1109/CAIDA51941.2021.9425290.

[29] R. Jodha and A. Dadheech, “Analysis and Evaluation of Unstructured data based on Stemming Algorithms,” *American International Journal of Research in Formal, Applied & Natural Sciences AIJRFANS*, pp. 19–201, 2019, [Online]. Available: http://www.iasir.net

[30] C. S. Saravana Kumar and R. Santhosh, “Effective information retrieval and feature minimization technique for semantic web data,” *Computers and Electrical Engineering*, vol. 81, Jan. 2020, doi: 10.1016/j.compeleceng.2019.106518.

[31] G. Attigen, M. Pai, and R. Pai, “Feature Selection Using Submodular Approach for Financial Big Data,” vol. 15, pp. 1320–1325, 2019.

[32] R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” 2011.

[33] A. G. Priya Varshini, K. Anitha Kumari, and V. Varadarajan, “Estimating software development efforts using a random forest-based stacked ensemble approach,” *Electronics (Switzerland)*, vol. 10, no. 10, May 2021, doi: 10.3390/electronics10101195.

[34] N. Rizun, A. Revina, and V. G. Meister, “Analyzing content of tasks in Business Process Management. Blending task execution and organization perspectives,” *Comput Ind*, vol. 130, Sep. 2021, doi: 10.1016/j.compind.2021.103463.

[35] J. P. Rajasingh, A. Jha, A. K. Singh, and S. Shekhar, “Boosting Tropical Cyclone Prediction Efficiency Through Time Series Analysis: An Ensemble Learning Approach,” 2021. [Online]. Available: http://www.paideumajournal.com

[36] H. NIGRO, “UNA REVISIÓN A LA MINERIA DE OPINIONES Y LOS RETOS DEL PNL,” *Revista Ingeniería, Matemáticas y Ciencias de la Información*, vol. 7, no. 13, pp. 105–110, Jan. 2020, doi: 10.21017/rimci.2020.v7.n13.a80.

[37] N. Díaz Roussel, M. J. Castro Bleda, and J. Á. González Barba, “Estudio comparativo de herramientas para tareas de Procesamiento de Lenguaje Natural,” 2019.

[38] J. Castillo *et al.*, “Desarrollo de Sistemas de Análisis de Texto,” 2017.

[39] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, “Natural language processing (NLP) in management research: A literature review,” *Journal of Management Analytics*, vol. 7, no. 2. Taylor and Francis Ltd., pp. 139–172, Apr. 02, 2020. doi: 10.1080/23270012.2020.1756939.

[40] M. M. Lopez and J. Kalita, “Deep Learning applied to NLP,” Mar. 2017, [Online]. Available: http://arxiv.org/abs/1703.03091

[41] J. ZhiXiang, Z. Chuang, X. Bo, and L. ZhiQing, “Research and implementation of intelligent chinese resume parsing,” in *Proceedings - 2009 WRI International Conference on Communications and Mobile Computing, CMC 2009*, 2009, vol. 3, pp. 588–593. doi: 10.1109/CMC.2009.253.

[42] K. Yu, G. Guan, and M. Zhou, “Resume Information Extraction with Cascaded Hybrid Model,” 2005.

[43] S. Pawar, R. Srivastava, and G. Keshav Palshikar, “Automatic Gazette Creation for Named Entity Recognition and Application to Resume Processing,” 2009, doi: 10.00.

[44] L. Sayfullina, E. Malmi, Y. Liao, and A. Jung, “Domain Adaptation for Resume Classification Using Convolutional Neural Networks,” Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.05576

[45] S. Maheshwary and H. Misra, “Matching Resumes to Jobs via Deep Siamese Network,” in *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, Apr. 2018, pp. 87–88. doi: 10.1145/3184558.3186942.

[46] B. Gaur, G. S. Saluja, H. B. Sivakumar, and S. Singh, “Semi-supervised deep learning based named entity recognition model to parse education section of resumes,” *Neural Comput Appl*, vol. 33, no. 11, pp. 5705–5718, Jun. 2021, doi: 10.1007/s00521-020-05351-2.

[47] R. Potolea, R. R. Slavescu, IEEE Romania Section, and Institute of Electrical and Electronics Engineers, *Proceedings, 2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing (ICCP) : Cluj-Napoca, Romania, September 7-9, 2017*.

[48] F. Dernoncourt, J. Y. Lee, and P. Szolovits, “NeuroNER: an easy-to-use program for named-entity recognition based on neural networks,” May 2017, [Online]. Available: http://arxiv.org/abs/1705.05487