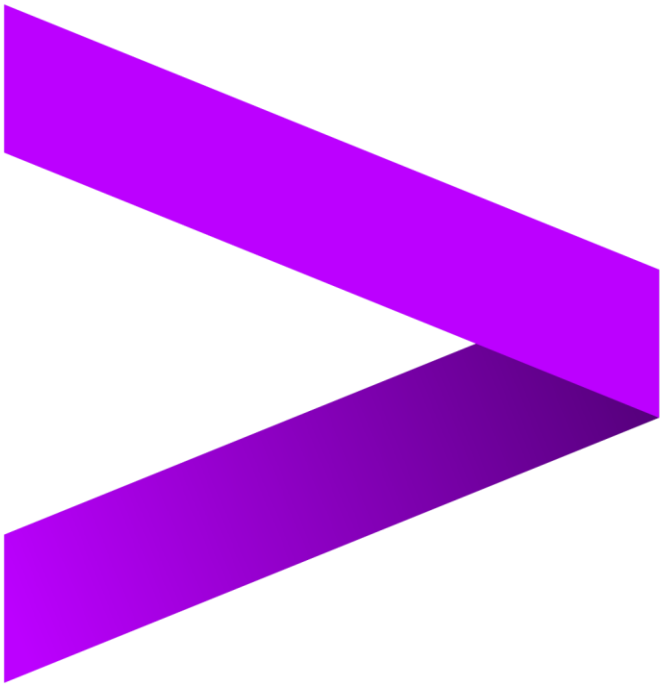
accenture

RETO ACCENTURE

NLP

Octubre 2021



1. EL RETO

El desarrollo de la Inteligencia Artificial para los juegos de mesa se viene produciendo con cada vez más naturalidad desde los últimos 20 años. En el año 1997, Gary Kasparov, entonces campeón mundial de ajedrez fue derrotado por Deep Blue, programa de inteligencia artificial desarrollado por IBM. También fue el caso de AlphaGo, un algoritmo de DeepMind que en 2016 venció a Lee Sedol, campeón de go, un juego altamente estratégico que requiere de una capacidad de computación mucho mayor que el ajedrez. En 2017, un nuevo algoritmo de DeepMind, AlphaGo Zero, se convirtió en el mejor jugador de go del mundo al derrotar a AlphaGo. Pero, a diferencia de todos los campeones hasta la fecha, AlphaGo Zero alcanzó su maestría sin aprender de nadie. Partiendo solo de las reglas del juego, se entrenó jugando partidas contra sí mismo y en tan solo tres horas estuvo preparado jugar contra AlphaGo, al que venció por cien a cero en cien partidas. Más recientemente, en mayo de 2019, se ha establecido un nuevo hito histórico, de nuevo los investigadores de DeepMind han conseguido que sus agentes de IA sean capaces de jugar y ganar a los humanos en un juego, pero, en este caso, Quake III Arena Capture the Flag, un videojuego multijugador en línea, algo imposible hasta la fecha, y que implica cooperar y competir con otros jugadores.

Bajo este contexto, se propone un reto para el desarrollo de algoritmos de NLP con el objetivo de identificar entidades en uno de los textos más comunes en la Administración Pública, el BOE. ¿Serán capaces los participantes de desarrollar un algoritmo que ayude al entendimiento de estos documentos tan densos? ¿Quién será el mejor de todos ellos?

2. INTRODUCCIÓN A LOS ALGORITMOS DE NLP

Un problema de procesamiento del lenguaje natural (NLP) consiste en analizar, interpretar y clasificar el contenido de un documento con información no estructurada, es decir, que no está en formato regular. Para ello, hay diferentes alternativas desde la aplicación de expresiones regulares a la utilización de redes neuronales mediante soluciones de Deep Learning.

Ante un problema de modelización de NLP, desde un punto de vista técnico, se requiere de una serie de tratamientos de la información

2.1 Preprocesado del texto

El texto original debe ser procesado antes de poder ser utilizado por los algoritmos de clasificación. Este proceso se lleva a cabo mediante la ejecución de uno o varios pasos, ejecutados en cierto orden, donde la salida de cada fase es tomada como entrada por la siguiente. Esto constituye una pipeline de pre procesado de los textos. Las principales técnicas son:

- Limpieza y normalizado
- Tokenizado
- Eliminación de palabras vacías
- Lematización
- Stemming
- Generación de N-gramas

2.2 Vectorización de los datos

Los tokens resultantes de la fase de pre procesado no pueden ser usados directamente por los algoritmos de clasificación, en su lugar, deben ser transformados en una representación matemática, generalmente vectores numéricos.

- La técnica de vectorización más simple es la denominada **Bolsa de Palabras** (Bag of Words en inglés). Este modelo asume que cuanto más aparece un token en un documento, más representativo es de su significado. Una de las desventajas de este método es que al dar más importancia a palabras que aparecen con más frecuencia se pueden obtener representaciones que contienen poca información relevante acerca del contenido del texto.
- Una versión mejorada de la anterior técnica es el **modelo TF-IDF** (Frecuencia de Término – Frecuencia Inversa de Documento, del inglés Term Frequency – Inverse Document Frequency). Se reduce la

importancia de términos que aparecen muy a menudo en el conjunto de los textos, ya que se asume que si una palabra es muy frecuente no es relevante para clasificar los textos en categorías.

- Una de las principales desventajas de los dos métodos presentados es que los vectores generados no consiguen capturar el significado y relaciones entre palabras, por ejemplo, no distinguen homónimos y no tienen en cuenta el contexto. Existen técnicas de vectorizado alternativas más sofisticadas que permiten solventar este problema, como son **Word2Vec**, **Doc2Vect** o modelos basados en arquitectura de **Transformers** (BERT, XLNet, GPT-2, etc). El entrenamiento de estos modelos requiere cantidades de texto muy superiores a las disponibles para el desarrollo de este proyecto, pero existen modelos preentrenados en español que se pueden utilizar para aplicarlos a las necesidades concretas del proyecto mediante técnicas de Transferencia de Aprendizaje (Transfer Learning en inglés).

2.3 Entrenamiento de los algoritmos de clasificación

Una vez vectorizados los documentos, ya se pueden usar como entrada de los modelos de clasificación. Hay dos tipos de algoritmos de clasificación:

- **Supervisados:** requieren textos etiquetados. Algunos ejemplos de algoritmos supervisados que se propone utilizar son el Clasificador Bayesiano Ingenuo (Naive Bayes en inglés), Máquinas de Vector Soporte, clasificadores basados en árboles y modelos de redes neuronales. Para poder emplear algoritmos de clasificación supervisados **es necesario anotar los datos**, es decir, definir las posibles categorías en las que se quiere clasificar los textos y asignar cada documento a una categoría.
- **No supervisados:** no requieren datos etiquetados, por lo que no hay que asignar las categorías a predecir de antemano. Esta clase de algoritmos genera grupos y coloca juntos textos que comparten características similares. En este apartado se sugiere utilizar la Asignación Latente de Dirichlet o el modelo de K-medias. En ambos la salida del modelo es el conjunto de temas de todos los textos y los temas que se tratan en cada uno de ellos. La diferencia consiste en que el LDA puede asignar a cada texto varias temáticas a la vez, mientras que el K-medias asigna un único tema a cada documento. Estos algoritmos son más útiles para generar una propuesta de las categorías existentes en los datos que para realizar la clasificación.

3. MECÁNICA DEL PROBLEMA

Tal y como se ha comentado, el objetivo del reto es la identificación de ciertas entidades dentro de los Boletines Oficiales del Estado (BOE).

El algoritmo desarrollado por los participantes será probado en un conjunto de documentos aleatorios del BOE previamente seleccionados por el jurado y será evaluado en función del acierto en la identificación de entidades en cada uno de los documentos.

Además, el jurado evaluará el algoritmo en sí mismo, así como la documentación o memoria presentada que explique la solución diseñada. Ambos entregables, algoritmo y documentación, tendrán que atenerse a las consideraciones y requisitos expuestos en los epígrafes siguientes.

3.1 Dataset para entrenamiento

La construcción del Dataset de entrada para el entramiento del modelo será responsabilidad de los participantes. Para ello, deberán descargarse de las páginas oficiales los documentos BOE que consideren y transformarlos en un formato que les permita empezar con la modelización del problema. Se recomienda el uso de los documentos en PDF, ya que la evaluación utilizará este formato de documento.

Los documentos podrán descargarse en el enlace oficial del BOE (https://www.boe.es/buscar/boe.php), y en el buscador filtrar por departamento “Ministerio de Trabajo y Asuntos Sociales”. El número de documentos a extraer será a elección de los participantes, sugiriendo utilizar al menos 10. Se recomienda también utilizar documentos que contengan información relevante para la extracción de las entidades indicadas en la siguiente sección.

3.2 Entidades a extraer

Las entidades que identificar dentro los documentos serán:

- Referencias jurídicas: las referencias jurídicas mencionan la aplicación de una ley o de un artículo en concreto de una ley. Algunos ejemplos de referencias jurídicas son: "*artículo 129 de la Ley 39/2015*", "*Ley 39/2015*", "*Ley de Enjuiciamiento Civil*", "*artículo 129 de la presente ley*", etc.
- Cantidades monetarias: dentro de los BOEs se pueden encontrar cuantías económicas referentes a diversas temáticas. Algunos ejemplos de cuantías son: "*965 euros/mes*", "*32,17 euros/día*", "*treinta y siete euros*", "*32,17 €*", etc.

3.3 Input del script

Como se ha comentado anteriormente, el algoritmo desarrollado por los participantes se probará en un conjunto previamente seleccionado por el jurado, compuesto por documentos del BOE. El formato de estos documentos será PDF y serán los mismos para todos los concursantes.

Estos documentos estarán alojados en una carpeta, siendo cada uno de los BOE un documento PDF distinto. El script desarrollado y entregado por los participantes debe permitir el paso como parámetro de la ruta (absoluta o relativa) a la carpeta que contiene los documentos de prueba. Adicionalmente, se requiere un segundo parámetro con el que el jurado identificará los equipos.

3.4 Ejecución y características del script

El algoritmo presentado por los equipos deberá ser desarrollado en lenguaje de programación Python (versión 3.8), con extensión ".py" y cuyo nombre será SCRIPT. Este script SCRIPT.py deberá ser autoinstalable en caso de disponer de alguna dependencia, pues deberá contener todo lo necesario para que el programa funcione en cualquier ordenador con conexión a internet.

Las características del ordenador donde se realizarán las validaciones son las siguientes:

- OS Name: Microsoft Windows 10 Enterprise
- System: 64-bit operating system, x64-based processor
- Processor: Intel(R) Core(TM) i5-8365U CPU @ 1.60GHz 1.90 GHz
- Installed Physical Memory (RAM): 8,0 GB

La ejecución del SCRIPT.py sobre los BOE de prueba deberá generar un archivo de texto por cada documento BOE con nombre SOLUCION_<parámetro identificativo del equipo>_<nombre documento BOE>.txt. Este archivo contendrá por cada fila las entidades identificadas en el documento en el orden de aparación en el mismo (si una entidad se repite, deberá aparecer las veces necesarias).

El algoritmo entregado por lo participantes debe estar basado en técnicas Machine Learning o Deep Learning (como las expuestas más arriba) y no en métodos heurísticos con expresiones regulares. El uso de este tipo de métodos penalizará negativamente en la evaluación del jurado.

3.5 Entregables

Cada participante deberá hacer entrega del fichero SCRIPT.py y un documento explicativo de lo realizado. Esta memoria, podrá ser una presentación o documento de texto. Deberá contener, al menos:

- Metodología y estrategia desarrollada para la resolución del problema.
- Principales dificultades encontradas.
- Conocimientos adquiridos.
- Fuentes principales de consulta (webs de consulta, libros…).

3.6 Criterios de valoración

La valoración subjetiva se realizará a través de los siguientes elementos:

- Puntos obtenidos en cada uno de los documentos BOE de prueba en función de las entidades identificadas de forma correcta.
- Memoria sobre el trabajo realizado en la que se valorará la claridad, la calidad, la innovación, la originalidad y la metodología propuesta entre otros.

Se tendrá en cuenta de forma positiva la rápidez en la ejecución del algoritmo, pudiendo desestimarse las ejecuciones que superen los 15 minutos en total (incluyendo todos los documentos).

El jurado se reserva el derecho a desestimar las soluciones presentadas por aquellos participantes que no cumplan los requisitos exigidos en este documento.