Procesamiento de Documentos con Deep Learning *(Document Processing with Deep Learning)*

Jorge Martín Acosta[1]

Material original autorizado para su primera publicación en la revista Ciencia y Tecnología de la Facultad de Ingeniería de la Universidad de Palermo.
Campo temático: Ciencias de la Computación.
Recepción: 29/8/2023 | Aceptación: 9/11/2023.

Resumen

Todas las empresas cuentan con una gran cantidad de documentos con formato de texto libre donde se guardan datos útiles para las mismas. Extraer información de diversos documentos que cuentan con una determinada estructura es relativamente sencillo utilizando las herramientas adecuadas, porque la estructura misma nos dice donde se puede localizar determinado dato. Cuando los documentos en cuestión no cuentan con una estructura, o peor aún, cuando la estructura cambia para un mismo tipo de documento de una región a otra, o incluso, dentro de una misma región, se requieren técnicas más complejas que permitan analizar cada documento y extraer los datos necesarios de manera tal que se pueda sortear el obstáculo de la estructura.

Palabras claves: inteligencia artificial; procesamiento natural del lenguaje; modelado de tópicos; clasificación de textos.

[1] Transener S.A. jorge.martin.acosta@gmail.com

Abstract

All companies have a large number of documents in free text format where useful data is stored for them. Extracting information from various documents that have a certain structure is relatively easy using the appropriate tools, because the structure itself tells us where certain data can be located. When the documents in question do not have a structure, or even worse, when the structure changes for the same type of document from one region to another, or even within the same region, more complex techniques are required to analyze each document. and extract the necessary data in such a way that the obstacle of the structure can be circumvented.

Keywords: artificial intelligence; natural language processing; topic modeling; text classification

1. Introducción

Una empresa líder en el servicio público de transporte de energía eléctrica en extra alta tensión en la República Argentina cuenta entre sus activos principales con Transformadores, Reactores y Líneas. Cuenta con gran cantidad de documentos como resultado del análisis de fallas en los mismos.

Todos estos documentos relacionados a los análisis de fallas se encuentran en formato de texto libre. Debido a que la empresa cuenta con diferentes regiones y distritos, y teniendo en cuenta que los documentos contienen un análisis de causas, el formato entre los diferentes informes tampoco es uniforme.

Este trabajo apunta a realizar un análisis de documentos correspondientes a Informes de Reportes de Perturbaciones (IRP) de la empresa en el período 2017-2023, utilizando Procesamiento de Lenguaje Natural (NLP), y su adecuación, para obtener datos que permitan, en un futuro cercano, generar un modelo de Inteligencia Artificial (IA) que pueda asistir al proceso de Toma de Decisiones, con miras a la migración hacia una estrategia de mantenimiento predictivo de los activos de la empresa.

El objetivo del análisis de los documentos es encontrar los puntos comunes en las estructuras de los diferentes tipos de documentos (IAP Informe de Análisis Preliminar/IRP Informe Regional de Perturbaciones/IFL Informe de Falla de Línea) y encontrar, para cada uno de esos puntos, los tópicos más usados.

2. Marco Teórico

2.1 Procesamiento de Lenguaje Natural.

El procesamiento de lenguaje natural es una técnica de la IA donde el modelo aprende palabras en forma de números y su relación con otras palabras. El conjunto de las diferentes palabras se llama Corpus.

Supongamos que tuviéramos el texto “Mi perro come balanceado y mi gato come atún.”. El corpus del texto sería de 7 (siete) palabras: mi, perro, come, balanceado, y, gato, atún.

Básicamente el NLP crea el modelo en dos pasos. El primer paso, genera una matriz donde cada palabra se representa como un vector de 1(unos) y 0 (ceros). El largo del vector coincide con la cantidad de palabras diferentes que se van a procesar. Si tuviéramos un texto de 10 palabras diferentes, el vector tendría un largo de 10 números. Cada vector tiene un número 1 (uno) en diferentes posiciones y el resto son 0 (ceros). Los vectores de números se conocen como Bolsa de Palabras (Bag of Words - BOW).

Para nuestro texto de ejemplo la bolsa de palabras sería:

Tabla 1 - Bolsa de palabras.

mi	1	0	0	0	0	0	0
perro	0	1	0	0	0	0	0
come	0	0	1	0	0	0	0
balanceado	0	0	0	1	0	0	0
y	0	0	0	0	1	0	0
gato	0	0	0	0	0	1	0
atún	0	0	0	0	0	0	1

Tabla 1

El segundo paso es la aplicación de la técnica de la ventana deslizante que consiste en determinar cómo se relaciona cada palabra con las que se encuentran antes y después.

Si aplicamos una ventana deslizante de 5 unidades, se toman 2 (dos) palabras antes y 2 (dos) palabras después de cada palabra que se analiza.

mi perro come balanceado y mi gato come atún
mi perro come balanceado y mi gato come atún
mi perro come balanceado y mi gato come atún
mi **perro come balanceado y mi** gato come atún
mi perro **come balanceado y mi gato** come atún
mi perro come **balanceado y mi gato come** atún
mi perro come balanceado **y mi gato come atún**
mi perro come balanceado y **mi gato come atún**
mi perro come balanceado y mi **gato come atún**

Al recorrer todo el texto analizando palabra por palabra, es decir, vector por vector, se van cambiando los valores para reflejar la relación de cada palabra con sus vecinos.

Los modelos solo analizan números por esta razón es que las palabras se transforman en vectores.

Una vez que el modelo queda entrenado con el corpus y la relación entre las palabras, se podría deducir que existe una cercanía entre perro-gato similar a la cercanía entre balanceado-atún.

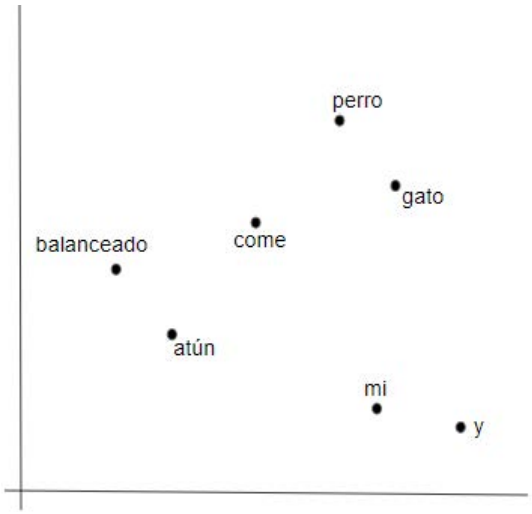

Figura 1 - Representación de los vectores en un eje cartesiano.

El modelo no sabe el significado de las palabras, no entiende que come un perro o que come un gato pero entiende que determinadas palabras por su posición y cercanía con otras tienen más importancia que otras.

2.2. Clasificación de textos no supervisada

La clasificación no supervisada de documentos es un modelo que trabaja el corpus de cada documento y le asigna una categoría (Garcia, 2020).

De entrada, antes de entrenar el modelo, se debe especificar la cantidad de categorías que debe tener el modelo. Si definimos que el modelo tendrá un máximo, por ejemplo de 3 categorías, el modelo mostrará las categorías 0;1 y 2.

A continuación, se clasifica cada texto dentro de las categorías sugeridas. Como resultado obtendremos, por ejemplo, teniendo una cantidad máxima de categorías de 3 (tres), que el texto "Texto 1" corresponde a la categoría 2, que el "Texto 2" corresponde a la categoría 0.

2.3. Modelado de Tópicos

En el modelado de tópicos, el modelo entrenado con un conjunto de textos al que llamamos corpus, establece para cada uno de los textos teniendo en cuenta el conjunto, una lista de palabras claves que permiten clasificar ese texto (*Introducción Al Topic Modeling Con Gensim (II): Asignación De Tópicos*, 2021).

Al configurar el proceso de modelado de tópicos se establece la cantidad máxima de clasificaciones.

De esta forma, para cada texto, obtendremos tópicos en el formato:
(ponderacion1 * palabra1; ponderacion2 * palabra2; ponderacionN * palabraN)
Con esta técnica de modelado de tópicos, obtenemos varios factores que nos indican porque el texto se clasifica de una forma determinada.

3. Desarrollo

3.1. Proceso de los Documentos

Todos los procesos se realizaron en lenguaje Python, en un equipo con 4 procesadores Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, 32 Gb de RAM, 100 GB de disco y con Sistema Operativo Windows 10 Pro de 64bits.

El entorno de desarrollo utilizado fue Jupyter y Spider del framework Anaconda.

Todos los documentos se encuentran organizados en carpetas con el formato {Año} y {Mes-Año} (Ejemplo 2022\02-2022 para los archivos de Febrero del Año 2022)

Dentro de las carpetas donde se encuentran almacenados los documentos, encontramos archivos en 3 (tres) formatos diferentes. Existen documentos en formato PDF, documentos en formato DOC y en formato DOCX. Para homogeneizar el proceso, lo primero que se hizo fue convertir todos los documentos al formato DOCX.

Para convertir documentos del formato PDF a DOCX se utiliza la librería pdf2docx.

Existen 1025 documentos para procesar desde 11-2017 hasta 07-2023.

Tabla 2 - Cantidad de documentos a procesar

Año	1	2	3	4	5	6	7	8	9	10	11	12	**Total**
2017											14	35	**49**
2018	38	9	19	7	11	12	10	30	24	9	18	23	**210**
2019	19	18	13	15	4	17	12	15	11	13	4	16	**157**
2020	7	19	5	7	10	8	15	31	23	17	8	11	**161**
2021	6	12	16	15	18	11	23	41	19	18	11	14	**204**
2022	37	10	11	9	4	8	9	18	17	18	6	13	**160**
2023	17	4	13	8	6	14	22						**84**
													1025

De esos 1025 documentos disponibles, solo se pudieron procesar 990 debido a que existen 35 documentos que no se pueden abrir por errores de formato. Estos archivos se dejaron de lado.

Tabla 3 - Cantidad de documentos procesados

Año	1	2	3	4	5	6	7	8	9	10	11	12	Total
2017											14	32	**46**
2018	37	9	18	7	11	12	10	30	24	9	17	22	**206**
2019	19	18	13	15	4	17	12	15	11	13	4	14	**155**
2020	7	19	5	7	10	8	15	31	23	17	8	9	**159**
2021	6	11	15	15	18	11	22	38	18	17	11	12	**194**
2022	37	10	9	9	4	8	8	18	17	18	6	13	**157**
2023	17	4	12	7	6	12	15						**73**
													990

De los 990 archivos que se pudieron procesar 38 se encontraban vacíos, con lo cual solo restaban 952 archivos para procesar.

Los archivos se procesan con un programa desarrollado en Python que lee cada párrafo del documento y lo agrega como un registro en una base de datos no estructurada (NoSQL) en MongoDb. Las tablas de los documentos las agrega en un único registro separando cada línea y cada columna.

Luego se procesa párrafo por párrafo juntando todo el contenido uno a continuación del otro.

3.2 Curado de los datos

Los datos registrados en los documentos analizados sufren de gran cantidad de irregularidades producto de su origen libre que se necesita corregir antes de poder realizar un procesamiento con NLP (*Introducción Al Topic Modeling Con Gensim (I): Fundamentos Y Preprocesamiento De Textos*, 2021).

Entre las irregularidades registradas se encuentran:

- textos surgidos como consecuencia de la transformación del documento en formato doc en formato docx.
- textos surgidos como consecuencia de la transformación del documento docx en texto plano.
- textos en mayúsculas y minúsculas.
- signos de puntuación.
- verbos conjugados.
- palabras como pronombres o adverbios que no agregan valor conocidas como STOPWORDS.

3.3 Aplicación de Técnicas de Procesamiento Natural del Lenguaje

De entrada, se sabe que existen 3 tipos de documentos: IAP, IRP e IFL entre los 952 documentos existentes. Por esta razón para clasificar los documentos se utilizaron 4 categorías.

Aplicando una clasificación no supervisada con un máximo de 4 grupos, el modelo clasificó los documentos de la siguiente manera: 225 IAP, 33 IRP, 672 IFL y 22 que no se pudieron clasificar.

La clasificación se realiza con la librería from sklearn.cluster utilizando el método KMeans.

Código de Implementación en Python:

kmeans = KMeans(n_clusters=cantidad_grupos, random_state = 0). fit (Matriz_texto)

Se procesan los documentos en formato DOCX, utilizando la librería docx2python.

Esta librería nos permite acceder al contenido de cada documento.

Se accede a cada párrafo del documento y se obtiene la lista de títulos de cada documento.

A continuación, se muestra la cantidad de ocurrencias de los títulos por cada tipo de documento:

Tabla 4 - Categorías de documentos IAP

Sección	Cantidad
Causas	**100**
Configuración Postfalla	**100**
Descripción cronológica de eventos	**100**
Informe	**100**
Normalización	**100**
Configuración Prefalla	**95**
Pérdida	**95**
Potencia Cortada (MW)	**90**
E.N.S. (MWh)	**70**
Medidas a adoptar	15
Aporte	10
Perturbación	5
Según el Libro de Novedades del COC se tiene:	5

Tabla 5 - Categorías de documentos IFL

Sección	Cantidad
Otros Comentarios	731
Distancia de Falla E.T. [Km.] – Según Localizadores	682
Datos Falla	674
Corresponde Progresiva	670
Detalles Falla Encontrada	669
Detección Falla en Torre Vano N° / Tipo / Fase	667
Observaciones Según Recorrido para Inspección - Causas/Efectos Falla:	666
Registros Fotográficos	650
Condiciones Meteorológicas	628
Motivo de la Falla (Para carga en BOE)	627
RECALCULO PC y C	38

Tabla 6 - Categorías de documentos IRP

Sección	Cantidad
ANALISIS DE LAS ACTUACIONES	29
ANEXOS	11
REFERENCIAS	9
DESCRIPCION DE LOS SUCESOS EN ORDEN CRONOLÓGICO	7
Distancia de Falla E.T. (km) – Según Localizadores:	7
IDENTIFICACIÓN DE CAUSAS	7
Comentario	6
CARACTERIZACIÓN DE LA ACTUACIÓN DE PROTECCIONES	5
Detalles de la Falla Encontrada:	5
MEDIDAS ADOPTADAS Y/O A ADOPTAR	5
Registros Fotográficos	5
RESUMEN DE LAS CAUSAS PROBABLES	5
SINTESIS	5
1° evento	3
2° evento	3
Condiciones Meteorológicas	3
3er Evento	2
Conclusiones y Recomendaciones	2
Corresponde Progresiva:	2

Para finalizar se toma el contenido que depende de cada título y se procesa mediante la técnica de modelado de tópicos.

Con este proceso se obtienen las palabras más representativas de cada parte del documento obteniendo una clasificación de los contenidos más comunes.

La obtención de tópicos se realiza con el método LdaModel de la librería gensim.models.

Código de Implementación en Python:

```
lda = LdaModel (corpus=corpus, id2word=diccionario,
 num_topics=cantidad_topicos, random_state=42,
 chunksize=1000, passes=20, alpha='auto')
```

4. Conclusiones

La clasificación de documentos como todo proceso no es exacta. Esos documentos que se clasificaron mal, agregaron ruido a la clasificación de títulos. Por ese motivo sólo se trabajó con aquellos títulos que eran más representativos.

En los documentos de tipo IAP e IFL, los resultados obtenidos, fueron concluyentes debido a que la muestra de documentos era lo suficientemente representativa como para obtener valores significativos. En cuanto a los documentos de tipo IRP, teniendo en cuenta que su muestra no es representativa, sumado al ruido agregado por las clasificaciones erróneas, los resultados tienen una dispersión tan grande que no permiten obtener conclusiones.

Referencias

Garcia, E. (2020, May 1). *Text Clustering. Este es uno de los temas más… | by Erick Garcia Ortiz*. Medium. Recuperado 27 de agosto de 2023, desde https://medium.com/@egocv/text-clustering-cdb6515bdc52

Introducción al topic modeling con Gensim (I): fundamentos y preprocesamiento de textos. (2021, March 18). Divulgando Machine Learning - El mundo de los datos. Recuperado 27 de agosto de 2023, desde https://elmundodelosdatos.com/topic-modeling-gensim-fundamentos-preprocesamiento-textos/

Introducción al topic modeling con Gensim (II): asignación de tópicos. (2021, March 31). Divulgando Machine Learning - El mundo de los datos. Recuperado 27 de agosto de 2023, desde https://elmundodelosdatos.com/topic-modeling-gensim-asignacion-topicos/