



GRADO EN INGENIERÍA EN  
TECNOLOGÍAS DE TELECOMUNICACIÓN

**TRABAJO FIN DE GRADO**

*ASOCIACIÓN DE TEXTOS MEDIANTE  
PROCESAMIENTO DEL LENGUAJE NATURAL  
(NLP)*

*Autor: Nicolás Corsini Santolaria*

*Director: José Antonio Ces Franjo*

Madrid, 2022-2023



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título *Asociación de textos mediante procesamiento del lenguaje natural (NLP)*, en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2022/23 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

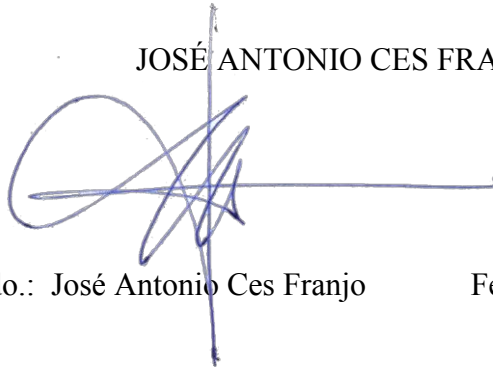
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Nicolás Corsini      Fecha: 01/ 06/ 2023



Autorizada la entrega del proyecto

JOSÉ ANTONIO CES FRANJO



Fdo.: José Antonio Ces Franjo      Fecha: 01/ 06/ 2023





GRADO EN INGENIERÍA EN  
TECNOLOGÍAS DE TELECOMUNICACIÓN

**TRABAJO FIN DE GRADO**

*ASOCIACIÓN DE TEXTOS MEDIANTE  
PROCESAMIENTO DEL LENGUAJE NATURAL  
(NLP)*

*Autor: Nicolás Corsini Santolaria*

*Director: José Antonio Ces Franjo*

Madrid, 2022-2023





# ASOCIACIÓN DE TEXTOS USANDO NLP

**Autor: Corsini Santolaria, Nicolás.**

Director: Ces Franjo, José Antonio.

Entidad Colaboradora: The Wise Seeker

## RESUMEN DEL PROYECTO

**Palabras clave:** NLP, Inteligencia Artificial, Embeddings, OpenAI

### 1. Introducción

En la actualidad, la cantidad de información digital que debe procesarse está en pleno crecimiento, por lo que se requiere metodologías eficientes y efectivas para su análisis. Ante esto, la inteligencia artificial ofrece soluciones rápidas y precisas para abordar problemas complejos aprovechando la capacidad computacional de los equipos modernos.

Uno de los numerosos ámbitos en los que se puede aplicar este tipo de tecnologías es el del lenguaje natural. Aunque un programa de inteligencia artificial no sea capaz de entender las palabras humanas, sí que puede simular su comprensión y realizar tareas de análisis sobre el lenguaje. Por tanto, estas herramientas permiten el procesamiento de textos de manera automática y a velocidades que ningún humano podría alcanzar, por lo que son realmente útiles en un mundo tan digitalizado.

Por todo esto, este trabajo de fin de grado se dedica a la cuestión del procesamiento del lenguaje natural. Concretamente, se centra en la asociación de textos de distinto formato y contenido, pero que mantengan un tema común.

### 2. Definición del proyecto

Este proyecto consiste en el desarrollo de un modelo de inteligencia artificial que permita generar asociaciones entre textos de distinto formato y contenido empleando técnicas de procesamiento del lenguaje natural (NLP). El sistema diseñado debe poder relacionar textos que compartan un mismo tema de manera automática y sin requerir de ningún tipo de entrenamiento. Además, este trabajo busca cubrir las necesidades de un proyecto real de la empresa colaboradora *The Wise Seeker*, en el que se desea realizar una recomendación de cursos online a partir de una evaluación sobre un determinado conocimiento.

Para llevar a cabo el proyecto, se dividen las tareas en tres fases principales.

En primer lugar, se hace un análisis del estado de la cuestión. En esta fase, se observan las tecnologías más relevantes dentro del mundo del procesamiento del lenguaje natural, con el fin de detectar cuáles podrían ser útiles para la generación de asociaciones.

En segundo lugar, se realizan pruebas prácticas sobre las herramientas analizadas en la fase anterior. El objetivo es hacer una selección de aquellas tecnologías más adecuadas, que permitan desarrollar un modelo que cumpla con los requisitos.



En tercer lugar, se lleva a cabo la fase de desarrollo del modelo. En esta etapa, se diseñan cada uno de los bloques del sistema, así como las conexiones entre ellos. Finalmente, se somete el modelo a pruebas con diferentes textos y se analizan los resultados.

### 3. Descripción del modelo

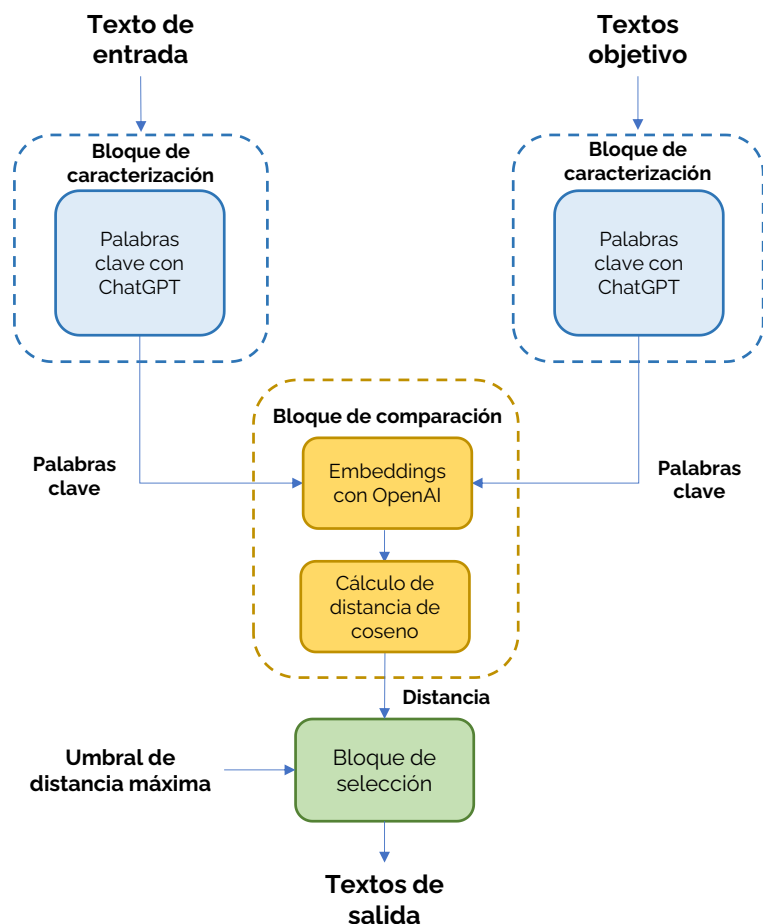
El modelo diseñado consta de tres bloques diferenciados.

El primer bloque es el de caracterización. Este recibe los textos y los manda a la API de ChatGPT para obtener unas palabras clave que definen su tema principal.

El segundo bloque es el de comparación, que recibe las caracterizaciones del primer bloque y las compara con el fin de generar asociaciones entre aquellos textos que comparten un mismo tema. Para hacer esto, primero transforma las palabras clave de cada elemento en vectores numéricos (embeddings) empleando la tecnología de OpenAI. Después, utiliza el cálculo vectorial de la distancia de coseno para determinar la similitud entre los temas de cada uno de los textos.

El tercer bloque recibe las distancias calculadas en el bloque de comparación y selecciona aquellos elementos que se encuentran más cerca en el espacio vectorial. Este bloque devuelve los textos que tratan el tema más similar al del texto de entrada, es decir, la salida final del sistema.

El diagrama del modelo diseñado es el siguiente:



## 4. Resultados

Algunos de los resultados obtenidos se muestran en las siguientes tablas. Encima de cada tabla se ve el título del texto introducido en el sistema, y en las columnas se muestran los textos recomendados.

### Routing

	course	distance
0	Cisco Curso de direccionamiento IP y Subnetting - CCNA	0.115
1	Cisco CCNA Fundamentos de Networking para Redes IP	0.122
2	Fundamentos Cisco Networking Parte 1	0.125
3	Cisco CCNA 200-301 en español	0.128
4	Cisco CCNA 200-301 - Practicas de configuracion en español	0.132
5	Como configurar una red de datos desde Cero de forma fácil	0.134
6	Certificación CCNA Versión 200-301. Aprende CCNA 2 y 3.	0.134
7	Cisco CCNA 200-301 en Español + Simulador de Preguntas !	0.137
8	Curso Cisco BGP nivel CCNP Encor by SeaCCNA	0.139
9	Fundamentos de Redes. Como se realizan las Comunicaciones.-	0.141

### Mercados Financieros

	course	distance
0	Inversión en Acciones y Bolsa de Valores (De 0 a Avanzado)	0.098
1	Valoración de Empresas: Descubre Cuánto Pagar por un Negocio	0.1
2	Modelos Financieros en Excel para la Valoración de Empresas	0.105
3	Finanzas Basicas: Aprende usando Excel	0.111
4	Finanzas para No Financieros	0.112
5	Análisis financiero corporativo	0.126
6	Domina las Matemáticas Financieras	0.128
7	Inteligencia Financiera: Guía completa Finanzas Personales	0.131
8	Curso de ANÁLISIS TÉCNICO Bursatil de Intermedio a AVANZADO	0.133
9	Introducción a las Finanzas	0.135

## Técnicas de Atención al Cliente

	course	distance
0	Calidad en Atención al Cliente	0.085
1	Habilidades Comunicativas: Mejora Tu Comunicación	0.097
2	Inteligencia Emocional para la Atención al Cliente.	0.104
3	Las Reglas de Oro de Atención a Clientes	0.104
4	Cómo Manejar Clientes Molestos	0.105
5	Habilidades Comunicativas (Guía Completa)	0.108
6	Comunicación Avanzada: Escucha Activa, Empatía y Asertividad	0.109
7	Comunicador@ Superestrella	0.111
8	Comunicación Avanzada / Feedback / Comentarios Constructivos	0.119
9	Habilidades de negociación y persuasión	0.12

Como se puede ver en todos los casos, el texto de entrada (cabecera de la tabla) trata un tema que está relacionado con el de los textos recomendados (primera columna).

### 5. Conclusiones

El sistema diseñado satisface los requisitos del proyecto. Es capaz de recibir un texto de entrada y compararlo con miles de otros textos para devolver aquellos que tratan un tema similar. Por ello, se cumple el objetivo del trabajo.

De cara al futuro, se pueden hacer mejoras como la implementación de un sistema de *feedback*, o la actualización a nuevas versiones de las tecnologías de *OpenAI*. También debe implementarse el modelo dentro de la plataforma de *The Wise Seeker* para terminar de probarlo en un entorno real.

### 6. Referencias

OpenAI, «Embeddings - OpenAI API,» [En línea]. Available: <https://platform.openai.com/docs/guides/embeddings>.

OpenAI, «Chat completion - OpenAI API,» [En línea]. Available: <https://platform.openai.com/docs/guides/chat>.

# TEXT ASSOCIATION USING NLP

**Author: Corsini Santolaria, Nicolás.**

Supervisor: Ces Franjón, José Antonio.

Collaborating Entity: The Wise Seeker

## PROJECT SUMMARY

**Key words:** NLP, Artificial Intelligence, Embeddings, OpenAI

### 7. Introduction

Currently, the amount of digital information that needs to be processed is continuously growing, requiring efficient and effective methodologies for analysis. In response to this, artificial intelligence provides fast and accurate solutions to tackle complex problems by leveraging the computational power of modern systems.

One of the many areas where this type of technology can be applied is natural language processing (NLP). Although an artificial intelligence program may not be able to understand human words, it can simulate comprehension and perform language analysis tasks. Therefore, these tools enable automatic text processing at speeds that no human could achieve, making them highly valuable in our digitized world.

Based on these factors, this project focuses on the field of natural language processing. Specifically, it centers around the association of texts with different formats and content, while maintaining a common theme.

### 8. Project description

This project involves the development of an artificial intelligence model that enables the generation of associations between texts of different formats and content using natural language processing (NLP) techniques. The designed system should be able to automatically relate texts that share a common theme without requiring any specific training. Additionally, this work aims to address the needs of a real project from the collaborating company, The Wise Seeker, where online course recommendations are desired based on an assessment of a specific knowledge.

To carry out the project, the tasks are divided into three main phases.

Firstly, a state-of-the-art analysis is conducted to explore the most relevant technologies in the field of natural language processing, in order to identify which ones could be useful for association generation.

Secondly, practical tests are performed on the tools analyzed in the previous phase. The goal is to select the most suitable technologies that allow the development of a model fulfilling the requirements.

Thirdly, the model development phase takes place. In this stage, each block of the system is designed along with the connections between them. Finally, the model is tested with different texts, and the results are analyzed.

## 9. Model description

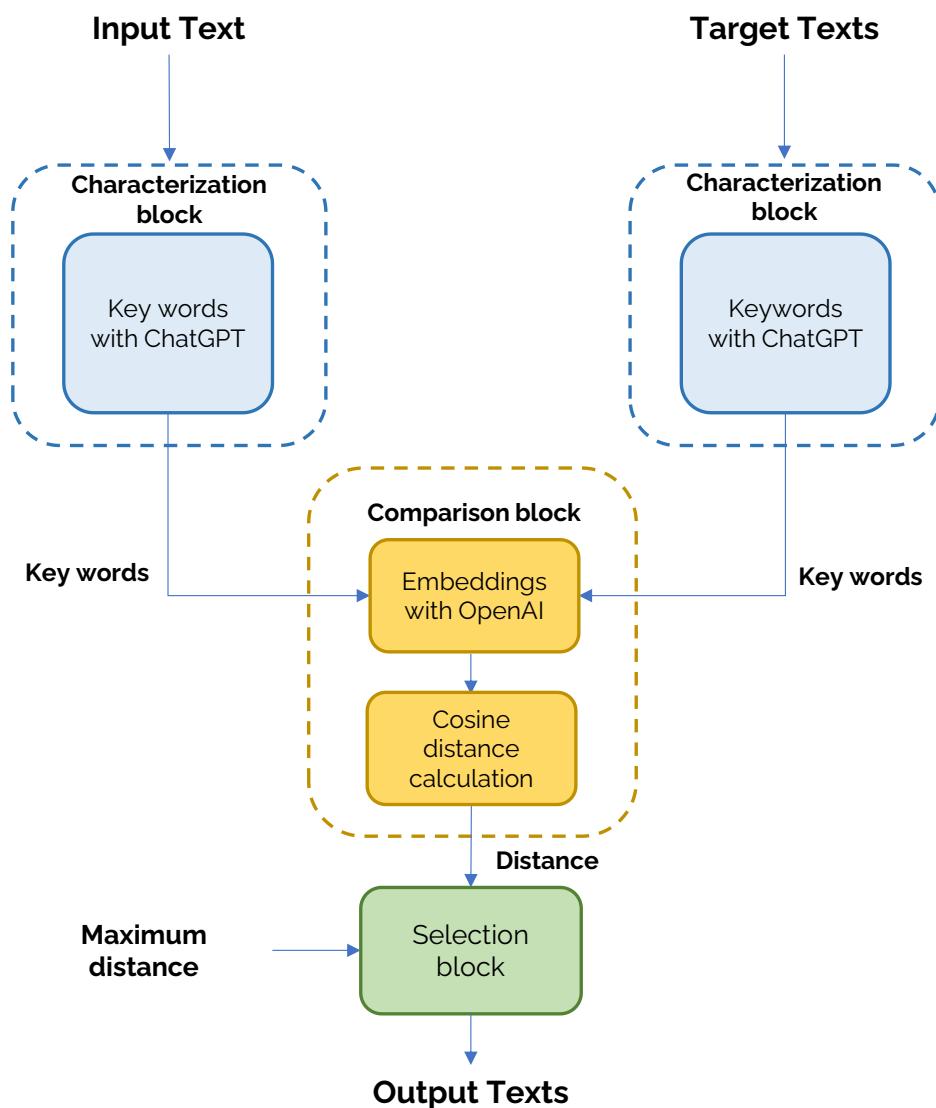
The designed model consists of three distinct blocks.

The first block is the characterization block. It receives the texts and sends them to the ChatGPT API to obtain keywords that define their main topic.

The second block is the comparison block, which receives the characterizations from the first block and compares them in order to generate associations between texts that share the same theme. To do this, it first transforms the keywords of each item into numerical vectors (embeddings) using OpenAI's technology. Then, it utilizes cosine distance vector calculation to determine the similarity between the themes of each text.

The third block receives the calculated distances from the comparison block and selects the elements that are closer in the vector space. This block returns the texts that address the most similar topic to the input text, which represents the final output of the system.

The diagram of the designed model is as follows:



## 10. Results

Some of the results obtained are shown in the following tables. The title of the input text is displayed above each table, and the recommended texts are shown in the columns.

### Routing

	course	distance
0	Cisco - TCP/IP & OSI Network Architecture Models	0.082
1	The World of Computer Networking. Your CCNA starts here	0.1
2	Networking Concepts with Socket Programming - Academic Level	0.101
3	IPv6 Internetworking Masterclass - Beginner to Advanced	0.105
4	Cisco BGP Configuration & Labs 2023- Basic to Advanced!	0.105
5	Introduction to IP Addressing and Subnetting the Easy Way	0.105
6	Introduction to Networking 2 Hour Crash Course	0.107
7	Computer Network: Networking fundamentals + Wireshark Basics	0.109
8	Cisco BGP (Border Gateway Protocol) Training	0.113
9	Cisco Certified Technician R&S RSTECH (100-490) Training	0.116

### Mercados Financieros

	course	distance
0	Financial markets: what beginners need to know!	0.071
1	Finance & Accounting Masterclass: Invest+Forecast+Value+More	0.086
2	The Complete Guide to the Global Capital Markets	0.089
3	Comprehensive Guide to Financial Markets, Investing &Trading	0.091
4	Introduction to Capital Markets	0.096
5	Build Financial Models & Value Companies The Easy Way	0.098
6	The Complete Financial Analyst Training & Investing Course	0.099
7	Understand Banks & Financial Markets	0.101
8	Capital Markets Immersion: A Financial Markets Introduction	0.105
9	Stock Market Trading & Investing: 8 Courses In 1 Bundle!	0.105

## Técnicas de Atención al Cliente

	course	distance
0	Customer Service: Soft Skills Fundamentals	0.09
1	Customer Service English Essentials	0.108
2	The Art of Negotiation - Become a Master Negotiator	0.11
3	Active Listening Skills: Give And Receive Feedback At Work	0.111
4	Customer Service Success: Take Your Skills to the Next Level	0.112
5	Increase Your Empathy Learn How to Communicate with Empathy	0.112
6	Communicating with Empathy	0.113
7	Brilliant Customer Service: How to Impress your Customers!	0.114
8	Customer Success: Working with Upset Customers	0.116
9	Business Communication and Ethics in Organizations -2023	0.119

As you it can be seen in all cases, the input text (table header) addresses a topic that is related to the recommended texts (first column).

### 11. Conclusions

The designed system meets the requirements of the project. It is capable of receiving an input text and comparing it with thousands of other texts to return those that address a similar topic. Thus, the objective of the project is fulfilled.

Looking ahead, there are several potential improvements that can be made. One possible enhancement is the implementation of a feedback system to gather user input and refine the recommendations over time. Additionally, updating the system to newer versions of OpenAI technologies can leverage any advancements and improvements in natural language processing.

Furthermore, it is essential to integrate and deploy the model within The Wise Seeker platform to conduct further testing and evaluation in a real-world environment. This will provide valuable insights and ensure the system's effectiveness and suitability for practical use.

### 12. References

OpenAI, «Embeddings - OpenAI API,» [Online]. Available: <https://platform.openai.com/docs/guides/embeddings>.

OpenAI, «Chat completion - OpenAI API,» [Online]. Available: <https://platform.openai.com/docs/guides/chat>.

Text translated to English with the help of ChatGPT.





## *Índice de la memoria*

<b>Capítulo 1. Introducción .....</b>	<b>6</b>
1.1 Motivación del proyecto.....	7
<b>Capítulo 2. Descripción de las Tecnologías.....</b>	<b>9</b>
2.1 Embeddings.....	9
2.2 Distancia de Coseno .....	13
<b>Capítulo 3. Estado de la Cuestión .....</b>	<b>17</b>
<b>Capítulo 4. Definición del Trabajo .....</b>	<b>22</b>
4.1 Justificación.....	22
4.1.1 Punto 1 .....	22
4.1.2 Punto 2 .....	23
4.1.3 Punto 3 .....	24
4.2 Objetivos .....	26
4.2.1 Primer objetivo.....	26
4.2.2 Segundo objetivo .....	26
4.2.3 Tercer objetivo.....	27
4.3 Metodología.....	27
4.3.1 Fase de análisis .....	27
4.3.2 Fase de selección.....	27
4.3.3 Fase de desarrollo.....	27
4.4 Planificación y Estimación Económica.....	28
4.4.1 Planificación.....	28
4.4.2 Estimación económica.....	29
<b>Capítulo 5. Desarrollo del proyecto .....</b>	<b>30</b>
5.1 Especificación de requisitos .....	30
5.1.1 Requisitos funcionales .....	30
5.1.2 Requisitos de estructura .....	31
5.1.3 Requisitos no funcionales .....	32
5.2 Fase de análisis del estado del arte.....	33

5.2.1 LDA .....	34
5.2.2 Spacy .....	34
5.2.3 Google Cloud Natural Language .....	35
5.2.4 Amazon Comprehend y Azure Cognitive Service .....	37
5.2.5 IBM Watson .....	37
5.2.6 OpenAI .....	39
5.2.7 Conclusiones del análisis del estado de la cuestión .....	40
5.3 Fase de selección de tecnologías .....	40
5.3.1 Contexto de las pruebas .....	40
5.3.2 Pruebas de caracterización .....	41
5.3.3 Pruebas de comparación .....	57
5.3.4 Conclusiones de la fase de selección de tecnologías .....	64
5.4 Fase de desarrollo del modelo .....	65
5.4.1 Modelo 1: IBM con Spacy .....	65
5.4.2 Modelo 2: ChatGPT con OpenAI .....	67
<b>Capítulo 6. Análisis de Resultados .....</b>	<b>71</b>
6.1 Resultados del modelo 1 .....	71
6.2 Resultados del modelo 2 .....	78
6.3 Conclusiones sobre los resultados .....	86
<b>Capítulo 7. Conclusiones y Trabajos Futuros .....</b>	<b>87</b>
7.1 Conclusiones .....	87
7.2 Trabajos futuros .....	88
<b>Capítulo 8. Bibliografía .....</b>	<b>89</b>
<b>ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS .....</b>	<b>91</b>
<b>ANEXO II: Repositorio en GitHub .....</b>	<b>92</b>

## *Índice de figuras*

Figura 1. Ejemplo de Embedding .....	10
Figura 2. Representación bidimensional de 12 frases. ....	12
Figura 3. Ejemplo de Distancia de Coseno con palabras .....	14
Figura 4. Ejemplo de Distancia de Coseno con frases. ....	15
Figura 5. Diagrama del algoritmo LDA. [1].....	18
Figura 6. Sistema de asociación con entrenamiento interno .....	24
Figura 7. Sistema de asociación con entrenamiento externo.....	25
Figura 8. Planificación del proyecto.....	29
Figura 9. Diagrama de bloques del sistema.....	32
Figura 10. Ejemplo de clasificación de textos con Google. [5].....	36
Figura 11. Ejemplo de clasificación de textos con IBM Watson [8] .....	38
Figura 12. Elementos a asociar.....	41
Figura 13. Problema de longitud de texto mínima con Google.....	46
Figura 14. Clasificación con IBM de un texto corto .....	51
Figura 15. Palabras clave de texto corto con ChatGPT.....	56
Figura 16. Representación de palabras clave en dos dimensiones con OpenAI .....	63
Figura 17. Diagrama del Modelo 1.....	66
Figura 18. Diagrama del Modelo 2.....	68
Figura 19. Resultados del Modelo 1 en el ámbito tecnológico .....	72
Figura 20. Resultados del Modelo 1 en el ámbito comercial .....	73
Figura 21. Resultados del Modelo 1 en el ámbito financiero .....	74
Figura 22. Resultados del Modelo 1 en el ámbito de RRHH .....	75
Figura 23. Resultados del Modelo 1 en el ámbito de Ingeniería .....	76
Figura 24. Resultados del Modelo 2 en el ámbito tecnológico. ....	79
Figura 25. Resultados del Modelo 2 en el ámbito comercial. ....	80
Figura 26. Resultados del Modelo 2 en el ámbito financiero.....	81

Figura 27. Resultados del Modelo 2 en el ámbito de RRHH. ....	82
Figura 28. Resultados del Modelo 2 en el ámbito de ingeniería. ....	83
Figura 29. Cursos recomendados para el bloque de finanzas.....	85
Figura 30. Página de ejecución del modelo.....	93

## *Índice de tablas*

Tabla 1. Clasificación con Google Cloud de los textos de Entrada .....	44
Tabla 2. Clasificación de textos objetivo .....	45
Tabla 3. Clasificación con IBM de los textos de entrada .....	49
Tabla 4. Clasificación con IBM de textos objetivo .....	49
Tabla 5. Palabras clave de textos de entrada con ChatGPT .....	54
Tabla 6. Palabras clave de textos objetivo con ChatGPT .....	55
Tabla 7. Similitud entre categorías de IBM con Spacy .....	59
Tabla 8. Similitud entre categorías de IBM con Spacy eliminando palabras en común. ....	60
Tabla 9. Distancias entre palabras clave con OpenAI .....	62
Tabla 10. Opciones de diseño del modelo final .....	64

## Capítulo 1. INTRODUCCIÓN

En la actualidad, con la expansión del mundo de la informática, la cantidad de información digital no para de crecer. Por ello, se buscan constantemente metodologías rápidas, precisas y eficientes de procesar estos datos. El uso de inteligencia artificial y *Machine Learning* permite, en ocasiones, acelerar el análisis de la información y obtener soluciones a problemas complejos aprovechándose de las capacidades computacionales de los equipos modernos. Es por ello que, hoy en día, la inteligencia artificial se aplica, de alguna forma, en prácticamente todos los sectores, dando soluciones a problemas que, de otra manera, serían muy tediosos de resolver.

Uno de los numerosos ámbitos en los que se puede aplicar la inteligencia artificial es el del lenguaje natural. A priori, el lenguaje parece algo ambiguo e irracional, por lo que sería difícil de analizar con una inteligencia artificial sin capacidad de comprensión. Sin embargo, lo cierto es que se puede extraer mucha información del lenguaje natural de las personas, ya sea escrito o hablado. Aunque una inteligencia artificial no pueda comprender e interpretar el sentido de las frases de manera natural, sí que puede hacerlo de manera simulada. En el lenguaje escrito, es capaz de leer un texto y analizar las palabras individualmente, así como la estructura de las frases. Los algoritmos más capaces pueden incluso simular la interpretación de un texto para obtener, por ejemplo, un resumen. Además, la verdadera fortaleza de este método de análisis reside en la velocidad, pues un proceso de este tipo analiza miles de textos en escasos minutos.

Por todo esto, se ha decidido dedicar este Proyecto de Fin de Grado precisamente a esta cuestión. Concretamente, estará centrado en la asociación de textos de cualquier fuente y tipo mediante las técnicas de inteligencia artificial que resulten ser más óptimas. Este proyecto es, por tanto, también un trabajo de investigación, en el que se analizarán las diferentes soluciones existentes y se buscará la mejor combinación de las mismas.

## 1.1 MOTIVACIÓN DEL PROYECTO

Hay varias razones que motivan la realización de este proyecto.

Por un lado, se desea indagar en el mundo de la inteligencia artificial y, concretamente, en el del procesamiento del lenguaje natural. Es simplemente de interés para el autor investigar y aprender sobre esta cuestión, sobre qué ofrecen los productos actuales y buscar la manera de implementarlos e incluso mejorarlos.

Por otro lado, se busca desarrollar una metodología de procesamiento de lenguaje natural que cubra las necesidades de un proyecto profesional real para la empresa *The Wise Seeker*. No solo consiste en un trabajo de interés propio, sino que tiene una aplicación real. *The Wise Seeker* posee una plataforma de evaluación y talento que basa su funcionamiento en la realización de evaluaciones de distinta índole, para validar objetivamente los conocimientos y habilidades de usuarios en la búsqueda de empleo y/o en su ejercicio profesional. Entonces, es de utilidad para la empresa tener una herramienta que analice, compare y filtre grandes cantidades de textos de manera automática. La motivación de este proyecto reside en que no existe una solución inmediata disponible para satisfacer esta necesidad y, por tanto, se debe llevar a cabo un trabajo de investigación y desarrollo.

Además, existe una motivación que va más allá del propio interés del autor, o del desarrollo de un proyecto de una empresa. Ésta proviene del deseo de contribuir en un ámbito que todavía está en una fase muy temprana. Más allá de la asociación de textos, el procesamiento del lenguaje natural mediante inteligencia artificial tiene infinidad de aplicaciones. En su esencia, el *NLP* marca el nivel de entendimiento entre el humano y la máquina, lo que es fundamental para cualquier sector en el que se implante algún tipo de inteligencia artificial. Actualmente, la interacción entre una persona y una inteligencia artificial es todavía muy mecánica, muy autómatas. En un mundo en el que se busca acercar las máquinas a los humanos mediante la simulación de una interacción de humano a humano, la capacidad de comprensión del lenguaje es clave, pero todavía está lejos de las sensaciones que se buscan.

Este proyecto, por tanto, busca hacer una aportación más dentro del desarrollo de una tecnología que muestra mucho potencial.

Finalmente, el procesamiento del lenguaje natural es una disciplina que se encuentra en un estado de máxima expansión. En el último año, han aparecido nuevas tecnologías que abren un nuevo frente de posibilidades. Ante la rápida evolución de estas herramientas, es normal que los procesos anteriormente diseñados queden obsoletos. Entonces, este proyecto viene motivado por la necesidad de crear una solución que contemple las herramientas más actualizadas del mercado.

En conclusión, las motivaciones de este proyecto son: la investigación y aprendizaje sobre el procesamiento del lenguaje natural, la cobertura de un proyecto empresarial, y la contribución al estado de la cuestión mediante una solución novedosa que implemente las últimas tecnologías.



## Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

El lenguaje de programación *Python* se ha usado para la plenitud de este proyecto, desde las funciones de tratamiento de datos hasta las llamadas a las diferentes *API*. Se ha trabajado en *Jupyter Notebook* y se han utilizado multitud de librerías como *pandas*. También, se ha usado *GitHub* para crear un repositorio con el objetivo de dejar ahí subido el proyecto. Estas tecnologías son muy generales, por lo que no requieren de explicación. Sin embargo, sí que hay dos conceptos que deben entenderse previamente para la comprensión del desarrollo del proyecto.

### 2.1 EMBEDDINGS

Dentro del contexto del *NLP*, la generación de *embeddings* es una etapa fundamental en el proceso de interpretación del lenguaje humano. Un *embedding* es, en esencia, una representación numérica de una palabra, que puede ser concebida como un vector dentro de un espacio vectorial. La manera de generar estos vectores puede variar según la tecnología, pero el concepto base es siempre el mismo. En general, se alimenta una red neuronal con grandes cantidades de textos reales que pueden ser de cualquier fuente, desde artículos de prensa o manuales de uso de herramientas, hasta novelas históricas y enciclopedias. Estos textos sirven como base para determinar la relación que existe entre todas las palabras del lenguaje. Con esto, el modelo comienza a asignar un vector a cada palabra, de tal manera que las palabras muy relacionadas tengan vectores más cercanos. Después, el modelo prueba a predecir palabras dentro de frases reales usando los vectores que ha generado. Esto lleva a un largo proceso de entrenamiento y ajuste de los *embeddings*, hasta que las predicciones resultan acertadas. Una vez ajustados los vectores, cada palabra queda con su valor numérico definitivo y permanente. Cuando el modelo recibe un texto de entrada, consulta los vectores correspondientes de cada palabra y los utiliza para realizar las tareas de *NLP*.

De esta manera, un modelo de inteligencia artificial tiene la capacidad de tratar las palabras como si fueran números, pudiendo entonces realizar operaciones matemáticas con ellas. Por ejemplo, los *embeddings* de dos palabras pueden compararse para estimar la similitud que existe entre ellas. Dos palabras muy similares en cuanto a significado tendrán *embeddings* más parecidos y, por tanto, sus vectores serán más próximos. Esta idea puede extrapolarse a frases completas, comparando los *embeddings* del conjunto de palabras que las forman.

En la *Figura 1* se muestra un ejemplo sencillo de *embedding* generado por la tecnología de *OpenAI* en *Python*:

```
word = "Dog"
embedding = embedding_from_string(word)
print(embedding)
[-0.0009602034115232527, -0.01513244491070509, -0.018407054245471954,
-0.02980327047407627, -0.017584795132279396, 0.012831563130021095, -0.
0051499358378350735, -0.028764627873897552, -0.017801178619265556, -0.
021580684930086136, 0.00946318730711937, 0.034996483474969864, 0.00048
2355710119009, -0.00718033779412508, -0.004518816247582436, 0.01741168
8342690468, 0.04711398109793663, 0.0035000089555978775, 0.014245270751
416683, -0.010819192975759506, -0.017931008711457253, 0.01142506767064
333, 0.019878463819622993, -0.011619813740253448, -0.01476459205150604
2, 0.006697080563753843, -0.004046378191560507, -0.019748633727431297,
-0.004096867982298136, -0.00495879678055644, 0.01115819439291954, -0.0
05932524334639311, -0.017830030992627144,
```

*Figura 1. Ejemplo de Embedding*

No se muestra todo el *embedding*, pues posee 1536 elementos. Como se puede ver, el *embedding* consiste en una lista de valores numéricos. Estos valores representan un vector único de 1536 dimensiones que identifica a la palabra “dog”.

A pesar de tener 1536 dimensiones, los *embeddings* pueden representarse en un espacio bidimensional usando el algoritmo de *t-SNE* (*t-Distributed Stochastic Neighbor Embedding*). Este consiste en tomar un conjunto de datos de alta dimensionalidad y transformarlo en un espacio de menor dimensión, de tal manera que se conserven las

distancias entre los puntos. Esto se logra calculando una distribución de probabilidad que modela las similitudes entre pares de puntos en los espacios de alta y baja dimensión. Después, se ajustan los parámetros de la distribución de probabilidad en el espacio de baja dimensión para que reflejen las similitudes entre los puntos del espacio de alta dimensión. De esta manera, minimizando la diferencia entre las distribuciones de probabilidad entre el espacio original y el espacio de menor dimensión, se pueden representar *embeddings* de 1536 dimensiones en un espacio bidimensional. En la *Figura 2* se muestra un ejemplo en el que se representan 12 frases de 3 temas diferentes:

- 4 frases sobre el clima:
  - *“Today is a sunny day.”*
  - *“It is very hot right now.”*
  - *“It might rain tomorrow.”*
  - *“The weather is very unpredictable.”*
- 3 frases sobre atletismo:
  - *“I am a very athletic person.”*
  - *“I like jogging.”*
  - *“The 100 meters sprint world record is 9.8 seconds by Usain Bolt.”*
- 5 frases sobre comida:
  - *“Pizza is an Italian dish.”*
  - *“I prefer meat over fish.”*
  - *“My favourite food is steak.”*
  - *“I don't like vegetables.”*
  - *“I don't usually order desert.”*

```
sentences = ["Today is a sunny day", "It is very hot right now", "It might rain tomorrow", "The weather  
is very unpredictable",  
  
            "I am a very athletic person", "I like jogging", "The 100 meters sprint world record is  
9.8 seconds by Usain Bolt",  
  
            "Pizza is an Italian dish", "I prefer meat over fish", "My favourite food is steak", "I  
don't like vegetables",  
            "I don't usually order desert."  
            ]  
  
embeddings = []  
  
# Calculate embeddings of sentences  
for sentence in sentences:  
    embeddings.append(embedding_from_string(sentence))  
  
# Convert to np array  
matrix = np.array(embeddings)  
  
# Create a t-SNE model and transform the data  
tsne = TSNE(n_components=2, perplexity=2, random_state=42, init='random', learning_rate=200)  
vis_dims = tsne.fit_transform(matrix)  
vis_dims.shape  
  
# Plot  
x = [x for x,y in vis_dims]  
y = [y for x,y in vis_dims]  
  
plt.scatter(x, y)  
plt.title("2-D representation of 12 sentences")
```

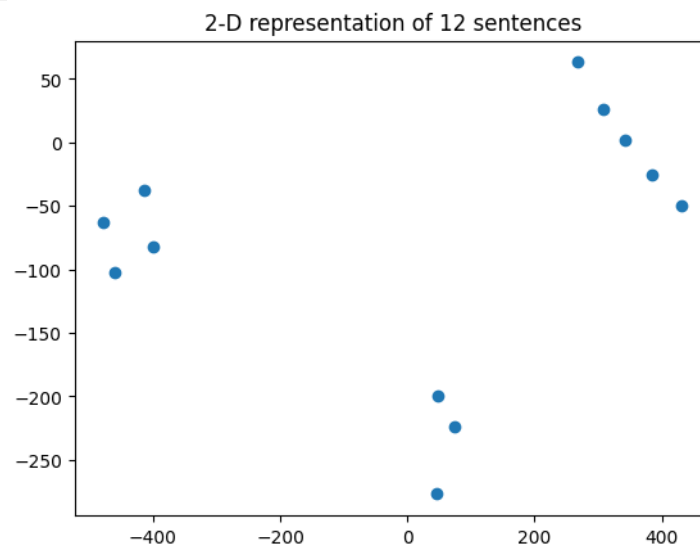


Figura 2. Representación bidimensional de 12 frases.

Como se puede observar, hay 3 grupos de puntos muy diferenciados. Estos coinciden con los temas de las 12 frases introducidas. Midiendo las distancias euclidianas entre los puntos en el espacio de 2 dimensiones, se puede obtener la similitud entre cada par de frases. Sin embargo, esta distancia no sería muy realista debido a la aproximación del algoritmo *t-SNE* por lo que, con una mayor cantidad de datos, se apreciarían imprecisiones. Por ello, el algoritmo *t-SNE* solo sirve para visualizar los *embeddings*. Las distancias deben medirse en el espacio de 1536 dimensiones con el fin de obtener los mejores resultados.

## 2.2 DISTANCIA DE COSENO

A la hora de comparar *embeddings*, se pueden emplear diferentes técnicas. Como, al fin y al cabo, los *embeddings* son vectores, se pueden aplicar los conceptos de medición de distancias de los espacios vectoriales. De esta manera, una de las medidas más utilizadas para calcular la distancia entre *embeddings* es la distancia de coseno.

La distancia de coseno es una medida de similitud entre dos vectores que se basa en el ángulo que forman en un espacio n-dimensional. La ecuación de la distancia de coseno entre dos vectores  $u$  y  $v$  es:

$$dist(u, v) = 1 - \frac{u \cdot v}{|u| \cdot |v|}$$

Donde  $u \cdot v$  es el producto escalar entre los vectores, y  $|u|$ ,  $|v|$  son las respectivas normas.

Así pues, en la *Figura 3* se muestra un ejemplo sencillo de esta medida calculando la distancia entre las palabras “Dog”, “Cat”, “Car”:

```
word_1 = "Dog"
u = embedding_from_string(word_1)

word_2 = "Cat"
v = embedding_from_string(word_2)

word_3 = "Car"
w = embedding_from_string(word_3)

distances = distances_from_embeddings(u, [u, v, w], distance_metric="cosine")

print("Distance between Dog and Dog: ", distances[0])
print("Distance between Dog and Cat: ", distances[1])
print("Distance between Dog and Car: ", distances[2])

Distance between Dog and Dog: 0
Distance between Dog and Cat: 0.12068051108681199
Distance between Dog and Car: 0.16764885517656636
```

*Figura 3. Ejemplo de Distancia de Coseno con palabras*

Como se puede observar, la distancia entre “Dog” y “Cat” es menor que la distancia entre “Dog” y “Car”. La distancia entre “Dog” y “Dog” es, lógicamente, nula.

Esta idea puede extrapolarse a frases completas. En la *Figura 4* se muestra un ejemplo, calculando la distancia entre las frases

- “Hoy hace un día soleado”
- “Me gusta salir a correr”
- “Hace mucho calor”
- “Soy una persona muy deportista”

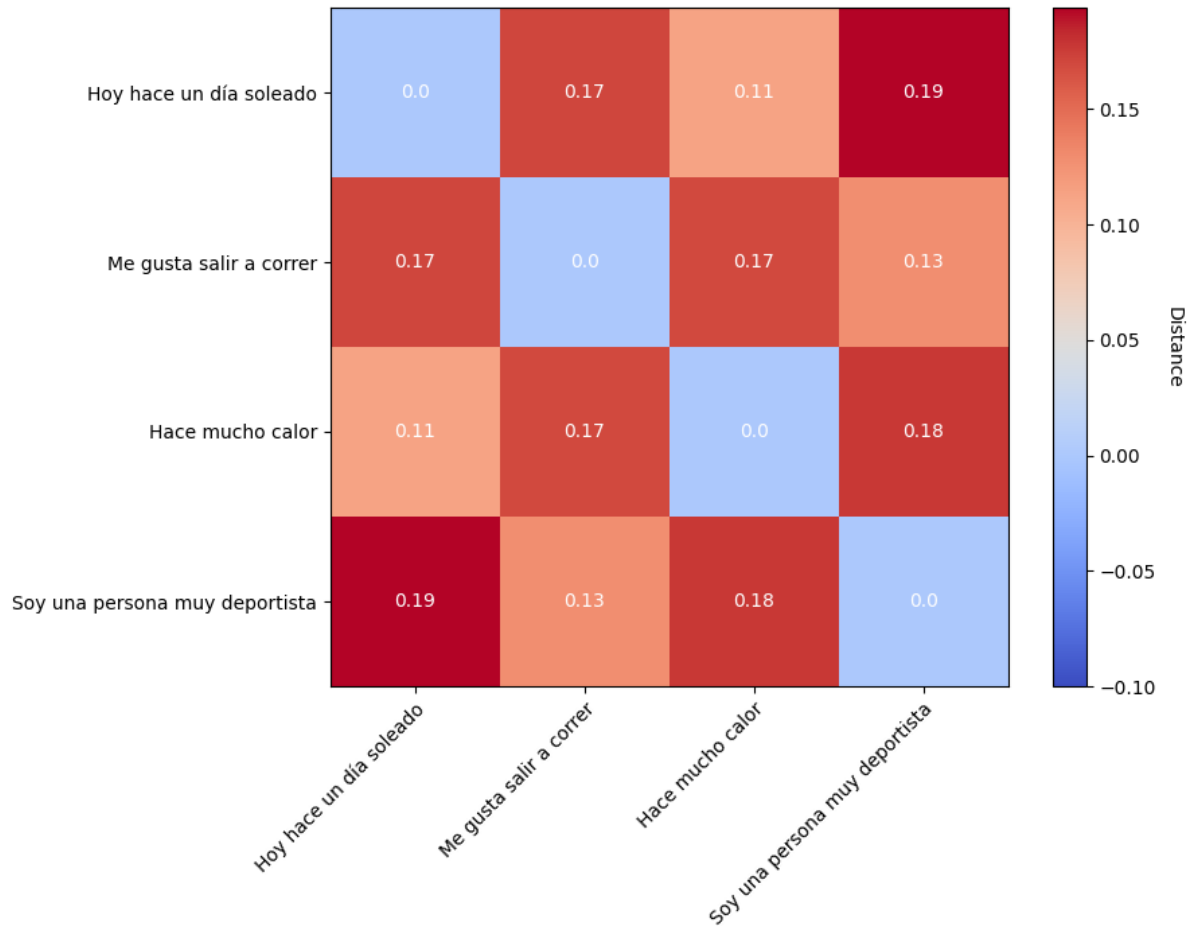


Figura 4. Ejemplo de Distancia de Coseno con frases.

Como se puede observar, la frase más cercana a “*Hoy hace un día soleado*” es “*Hace mucho calor*”, y la frase más cercana a “*Me gusta salir a correr*” es “*Soy una persona muy deportista*”. Además, se demuestra que el modelo generador de embeddings de *OpenAI* funciona también en otros lenguajes como el español.

Este ejemplo es una muestra del gran potencial que tienen los *embeddings* junto con la medida de la distancia de coseno. Combinando estas dos tecnologías, se puede cuantificar la similitud que existe entre dos palabras o dos frases, lo que es muy valioso dentro del mundo

del procesamiento del lenguaje natural. Dotando a una inteligencia artificial con estas técnicas, se le otorga la capacidad de relacionar conceptos, tal y como podría hacerlo una persona. Por todo ello, estas dos tecnologías construyen, sin duda, uno de los pilares del *NLP*.



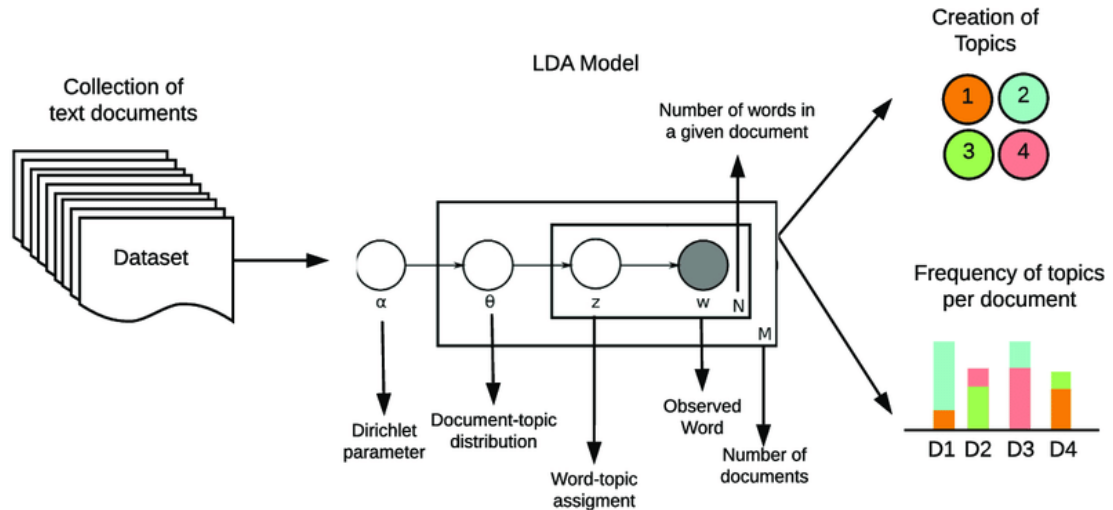
## Capítulo 3. ESTADO DE LA CUESTIÓN

Actualmente, existen muchas soluciones ante el análisis de textos mediante el procesamiento del lenguaje natural (*NLP*), y tienen infinidad de aplicaciones. El *NLP* se usa, por ejemplo, para analizar reseñas escritas por usuarios que evalúan un servicio, con el fin de determinar si son positivas o negativas. También se emplea para leer y sacar información básica de grandes cantidades de documentos como facturas y recibos en poco tiempo. Otro ejemplo de su aplicación es el reconocimiento de voz, que da pie a tecnologías como los robots de chat (*Siri* de *Apple*, *Alexa* de *Amazon*) o el dictado automático. Por otro lado, también se usa en todo lo relacionado con los sistemas de recomendaciones de, por ejemplo, películas, series, o cualquier producto *online*. En definitiva, como se puede ver, la lista de utilidades del *NLP* es casi interminable.

Los proveedores de estas aplicaciones son de fuentes muy diversas. Por un lado, existen los gigantes tecnológicos que ofrecen sus propias tecnologías a la venta mediante herramientas de Software o implementaciones a través de servicios *API REST*. Estos son, principalmente: *Google*, *IBM*, *Microsoft* y *AWS*. Por otro lado, existen miles de propuestas abiertas de usuarios particulares que usan algoritmos programables para crear sus propios modelos. Cada opción tiene sus ventajas, aunque, lógicamente, los servicios de las grandes empresas tecnológicas suelen ser los más probados y eficaces. Sin embargo, estos pueden sufrir de abarcar demasiadas funcionalidades cuando lo que se busca es una utilidad en concreto.

En cuanto al tema en particular de este trabajo, que es la asociación de textos, hay algunas soluciones ya existentes. En el ámbito de particulares, el algoritmo más usado entre los usuarios es, de lejos, el *Latent Dirichlet Allocation (LDA)*. Este algoritmo se basa en la idea de que todo documento puede describirse como una distribución probabilística de temas y que, a su vez, todo tema puede describirse como una distribución probabilística de palabras. De esta manera, el algoritmo crea un modelo estadístico de las palabras más relevantes del texto y genera las categorías a las que pertenece el documento. El *LDA* es no supervisado, por lo que las categorías resultantes no están previamente definidas. Esto aporta flexibilidad

y escalabilidad, pero puede restar precisión. En la *Figura 5* se muestra el funcionamiento del algoritmo LDA.



*Figura 5. Diagrama del algoritmo LDA. [1]*

Resumidamente, el LDA recorre cada palabra de un documento y le asigna una probabilidad de pertenecer a un tema en concreto. Al principio, los temas asignados son aleatorios, así que la probabilidad de que esa palabra pertenezca a ese tema es muy baja. Con cada iteración del algoritmo, la precisión del tema asignado a cada palabra aumenta, así como el nivel de confianza. Al final, se obtiene una distribución de los temas que aparecen en el documento con mayor frecuencia, pudiendo así generar una idea del tema general del mismo.

Además de algoritmos como el *LDA*, existen numerosas librerías abiertas en *Python* con herramientas de análisis de textos mediante *NLP* como, por ejemplo, *Spacy*. Con esta librería, se puede analizar la similitud entre palabras o frases mediante la generación de *embeddings* (ver *Descripción de las Tecnologías*, sección 2.1) basándose en un modelo gratuito ya entrenado. También posee funcionalidades de identificación de palabras clave dentro de una frase, siendo capaz de extraer los verbos, el sujeto o los complementos. Esta herramienta tiene la ventaja de que es totalmente gratuita y no requiere de ninguna *API*, pues toda la funcionalidad se encuentra dentro de su librería de *Python*.

Por otro lado, las soluciones de las grandes empresas tecnológicas ofrecen una inteligencia artificial ya programada capaz de analizar textos de muchas maneras. Estas opciones no solamente permiten realizar la clasificación de un texto, sino que también pueden llevar a cabo un análisis sintáctico, o un análisis de sentimiento, que busca detectar si el mensaje del texto es optimista o pesimista. El ejemplo más cercano al público es el de *Google Cloud*, que deja a disposición de sus usuarios una *API*, de pago con prueba gratuita, a la que se pueden hacer llamadas. En la petición de la llamada se incluye el texto que se quiere analizar, y en la respuesta viene el análisis de la inteligencia artificial que hay detrás. En cuanto a la asociación de textos, este servicio utiliza un algoritmo supervisado con más de 700 categorías definidas para clasificar un documento. Esta funcionalidad responde con las categorías que más concuerdan con el texto, junto con un coeficiente de confianza, que representa la certeza del resultado. En un principio, esta solución puede parecer acertada para caracterizar textos, pero puede mostrarse limitada por las categorías ya definidas. Con un texto que trata un tema muy concreto, la categoría resultante es una más general de lo que posiblemente se desea. A pesar de ello, es un buen acercamiento a la cuestión.

Otra de las grandes empresas que lideran la tecnología del procesamiento del lenguaje natural es *Amazon* con su servicio *Amazon Comprehend*. Este servicio ofrece funcionalidades parecidas a *Google Cloud*, pero con una aproximación más técnica. Por ejemplo, permite modificar los modelos que utiliza el algoritmo para ejecutar análisis de clasificación o de reconocimiento de entidades. Es decir, *Amazon Comprehend* da la opción de personalizar y entrenar los modelos con los datos que el usuario quiera. Con esto, se consiguen resultados más concretos y adecuados, a costa de la complejidad que requiere entrenar el algoritmo. Además de estas funciones personalizables, el servicio también cuenta con una tecnología de extracción de frases y palabras clave, que podría resultar muy útil para el objetivo de este proyecto. En definitiva, *Amazon Comprehend* aporta un punto de vista más complejo y con una interfaz más técnica que *Google Cloud*. Esto podría hacer que este servicio resultase más adecuado para el trabajo. Entonces, debe ser probado.

Por parte de *Microsoft*, *Azure* también dispone de una plataforma de *NLP*: *Azure Cognitive Service*. Al igual que las anteriores, este servicio funciona a base de llamadas a una *API*, que

puede cumplir con varias funciones. Puesto que no dispone de la capacidad de personalización de *Amazon Comprehend*, se acerca más a la alternativa de *Google*. Sin embargo, no posee la funcionalidad de clasificar textos en categorías. A cambio, es capaz de hacer un resumen de un documento, y se centra sobre todo en el reconocimiento de entidades con nombre como personas, eventos y organizadores. Aunque a priori parezca menos apropiado de cara a la caracterización de textos, también es conveniente estudiar este servicio.

Otra de las opciones más reconocidas dentro de este ámbito pertenece a la tecnología de *Watson Natural Language Understanding*, de *IBM*. A través de una *API*, este servicio ofrece posiblemente el acercamiento más completo a la cuestión. Combina la posibilidad de *AWS* de personalizar los modelos, con los análisis predefinidos y más simples de *Google*, pero más completos y variados. De esta manera, el servicio de *IBM* muestra tres funcionalidades principales a la hora de analizar un texto. La primera es la funcionalidad de extracción, que obtiene las entidades, los conceptos y las palabras clave del documento. La segunda es la de clasificación, que categoriza el texto en cuanto a sentimiento, emoción del mensaje, y temas que trata. La clasificación en temas parece ser muy similar a la de *Google Cloud*, pues usa las mismas 700 categorías predefinidas. La tercera funcionalidad que otorga este servicio es la que se encarga de hacer un análisis sintáctico y semántico. De cara a la asociación de textos, las dos funcionalidades más interesantes a priori son las de clasificación y la de extracción de conceptos y palabras clave. Puesto que esta tecnología engloba las ventajas de cada una de las anteriores, estudiarla es de vital importancia.

Por último, existe otra tecnología que recientemente ha causado grandes sensaciones en el mundo de la inteligencia artificial. *OpenAI* ha dado acceso público a su nueva tecnología más avanzada: *Chat-GPT*. Esta herramienta consiste en un modelo de lenguaje basado en la arquitectura *GPT* que es capaz de comprender y generar texto con una naturalidad jamás vista. Esta inteligencia artificial ha sido entrenada utilizando una enorme cantidad de textos de todo tipo de fuentes, por lo que puede responder todo tipo de preguntas. Además, es capaz de resolver problemas de lógica y de realizar multitud de tareas de procesamiento de lenguaje natural. Gracias a su buen rendimiento, *Chat-GPT* está empezando a implementarse para una

gran variedad de aplicaciones, incluyendo *chatbots*, asistentes virtuales, herramientas de escritura y mucho más, todo mediante el uso de la *API* de *OpenAI*. Por otro lado, *OpenAI* también ha actualizado su modelo generador de *embeddings* gracias a la tecnología *GPT*. Por todo esto, las tecnologías de esta empresa pueden resultar clave para este proyecto, pues son, posiblemente, las más avanzadas dentro del contexto del procesamiento del lenguaje natural.

## **Capítulo 4. DEFINICIÓN DEL TRABAJO**

### **4.1 JUSTIFICACIÓN**

La justificación de este proyecto debe ser clara y contundente, pues requiere de mucho tiempo de investigación y de desarrollo. Por tanto, a continuación se describen las tres razones principales que justifican su realización.

#### **4.1.1 PUNTO 1**

Viendo en la sección *Estado de la Cuestión* la cantidad de opciones que ya existen a la hora de analizar textos, cabe preguntarse si este proyecto es realmente necesario. Con todas las alternativas, tanto de grandes empresas como de pequeños desarrolladores, se podría pensar que al menos una de ellas debería ser capaz de cumplir con los objetivos de este proyecto. Sin embargo, tras analizar todas las tecnologías, se llega a la conclusión de que ninguna logra el resultado deseado por sí misma. Esto es, fundamentalmente, debido a que la idea de este proyecto es hacer un sistema altamente flexible que pueda asociar textos de cualquier tipo. El objetivo es que se introduzca en el sistema un texto que trate un tema en concreto, y que este devuelva una lista de textos que traten un tema similar al del primero. Esto conlleva dos etapas dentro del proceso. Primero debe caracterizarse de alguna manera el tema del texto introducido y, en segundo lugar, debe compararse ese tema con el tema de una pila de textos guardados en el sistema para devolver los más similares. Además, los textos a tratar no deben verse restringidos por ningún formato fijo. Es decir, el sistema debe ser capaz de tratar textos que describan cualquier tipo de contenido como, por ejemplo, artículos de prensa, descripción de películas o de cualquier producto, archivos tipo *JSON* que representen elementos como cursos online, etc. El límite solamente debe ser establecido por la longitud del texto introducido, pero nunca por su contenido o por su forma. Actualmente, no existe una tecnología que cubra estos requerimientos. Los servicios ya existentes aportan unas bases sólidas para el análisis y la comparativa de los textos, pero no ofrecen un proceso tan

completo. Este proyecto busca integrar la mejor combinación de estas soluciones para diseñar un sistema altamente flexible, preciso y sencillo de implementar.

#### **4.1.2 PUNTO 2**

Ya habiendo aclarado que no existe una solución actual al motivo del proyecto, es necesario cuestionarse si realmente la funcionalidad que se busca podría llegar a ser útil. Esta duda se resuelve rápidamente pensando en las aplicaciones que podría tener una tecnología como la que se busca desarrollar.

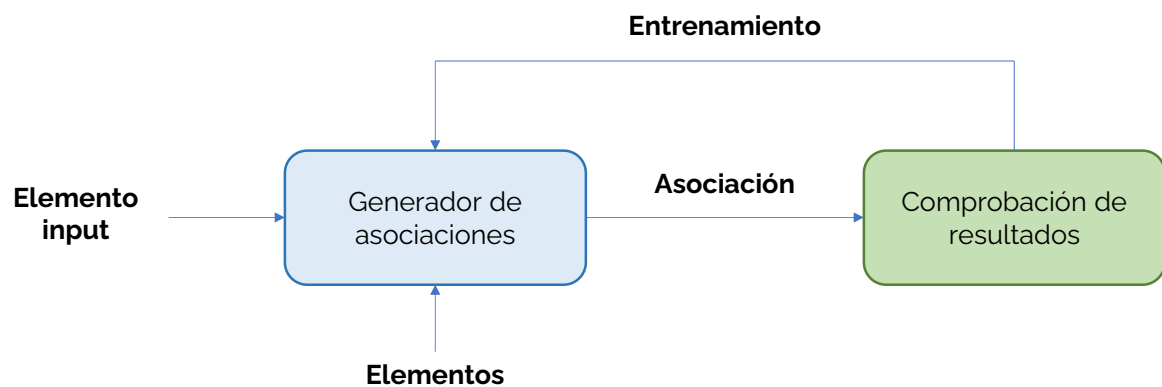
Una herramienta capaz de asociar textos que tengan algo en común es extremadamente útil en los sistemas de recomendación. Este tipo de sistemas se usan en toda clase de plataformas como plataformas de *streaming* de series y películas o tiendas online. Generalmente, las herramientas de recomendación ya existentes están diseñadas para trabajar con elementos concretos de la plataforma en cuestión, como películas, libros, productos, etc. Cada plataforma tiene su propio sistema de recomendación que funciona solamente aplicado a un tipo de elemento. Puede que estos programas cumplan su función, pero no emplean las últimas tecnologías de procesamiento de lenguaje natural y, por tanto, pueden ser mejorados.

Además, un sistema tan flexible en cuanto a la forma de los textos permite asociar elementos muy distintos. Es decir, dentro de una asociación de dos textos, uno puede tratarse de, por ejemplo, un examen tipo test sobre un tema en concreto, y el otro puede ser un curso online que trate ese mismo tema. Estos dos elementos son completamente diferentes en cuanto a forma, pero, al tratar un tema en común, se verían relacionados. Este ejemplo es el que se va a explorar a lo largo de todo el proyecto, pues demuestra el potencial de la idea por encima de los sistemas de recomendación convencionales, y no es una funcionalidad tan común como la recomendación de películas o series. Como se verá más adelante, aunque la idea es que la herramienta desarrollada sea aplicable a una gran cantidad de casos, el objetivo principal de este trabajo es dar soporte a la aplicación concreta de recomendación de cursos online.

### 4.1.3 PUNTO 3

Debido a las condiciones de desarrollo de este proyecto, no existe la posibilidad de diseñar un sistema de asociación de textos que requiera una fase de entrenamiento. A diferencia de muchas tecnologías de inteligencia artificial, esta deberá usar modelos de lenguaje ya entrenados. Esto puede verse como una desventaja, pues el objetivo del entrenamiento de un modelo es, generalmente, aumentar la precisión de los resultados. Sin embargo, la característica de no necesitar entrenamiento puede verse como una ventaja, siempre y cuando los resultados del modelo sean buenos. Al no requerir esta fase, la implementación es mucho más rápida y sencilla. En vez de tener que introducir miles de datos de prueba para entrenar el sistema, la idea es utilizar modelos de lenguaje ya entrenados por otras fuentes. Gracias a esto, la tecnología a desarrollar debe dar resultados satisfactorios desde el principio de su implementación, sin necesidad de entrenamiento.

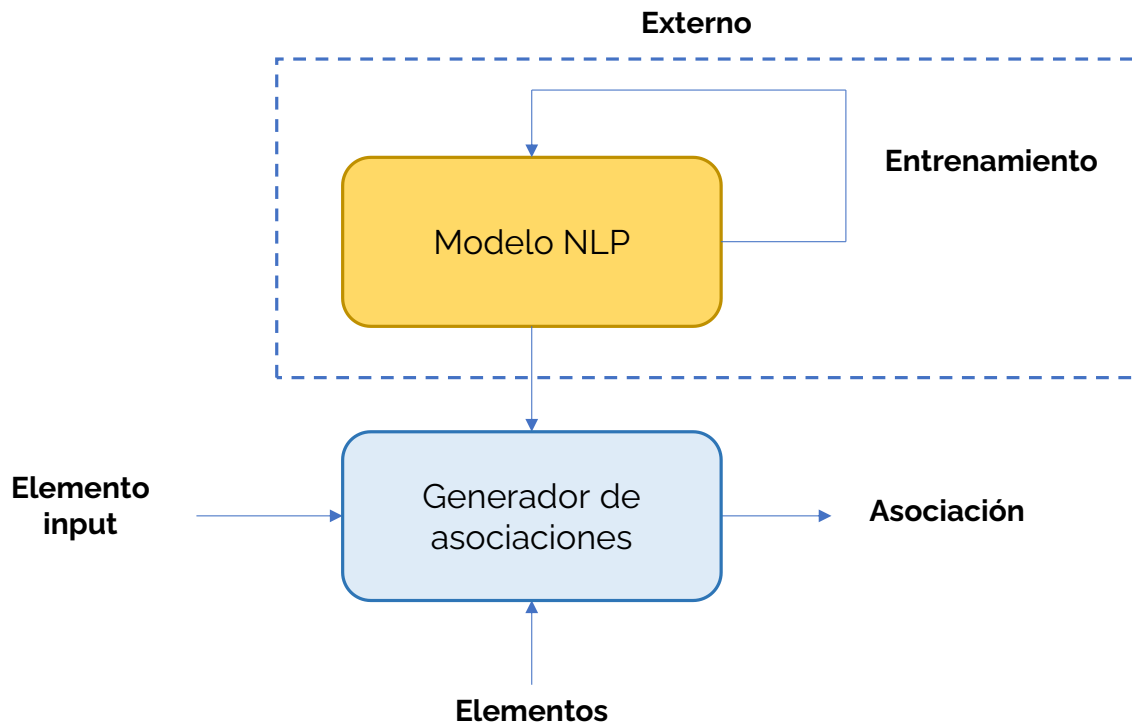
En la *Figura 6*, se muestra un esquema simplificado y genérico de un sistema de asociación de elementos con entrenamiento interno. Se introduce un *Elemento Input* como entrada, y el generador de asociaciones lo compara con el resto de elementos para obtener las relaciones. Como se puede ver, después se comprueban los resultados de la asociación, y se realimentan al modelo para entrenarlo e ir aumentando poco a poco la precisión. Para que los resultados finales sean buenos, hace falta un entrenamiento muy extenso y, sobre todo, muchos datos de asociaciones correctas para poder comprobar que las que genera el modelo son válidas.



*Figura 6. Sistema de asociación con entrenamiento interno*



Por otro lado, en la *Figura 7*, se muestra un esquema simplificado de un sistema de asociación con entrenamiento externo. Los elementos dentro del recuadro de línea discontinua corresponden con un modelo de procesamiento de lenguaje natural externo entrenado de forma independiente a la asociación de textos. De esta manera, se emplea una inteligencia artificial que sepa interpretar el lenguaje escrito en general y que, por tanto, sea capaz de interpretar los textos del sistema, así como de generar asociaciones entre ellos. Como el entrenamiento se hace sobre el modelo de *NLP* y no sobre el sistema completo de asociación, el formato de los textos es totalmente flexible.



*Figura 7. Sistema de asociación con entrenamiento externo.*

En definitiva, este proyecto busca generar una funcionalidad que no existe, empleando las últimas tecnologías y creando una herramienta con gran cantidad de aplicaciones. Si se desarrolla correctamente, la solución final puede ser de gran utilidad para cualquier programa que requiera asociar elementos de texto, sin que importe la procedencia ni el formato de los mismos.

## **4.2 OBJETIVOS**

Se definen tres objetivos principales que se persiguen con este proyecto.

### **4.2.1 PRIMER OBJETIVO**

El primer objetivo es el de estudiar y comprender todo el estado de la cuestión. Consiste en familiarizarse con los distintos servicios ya existentes, para entender qué ofrece cada uno. Se trata de indagar en las técnicas de procesamiento de lenguaje natural y hacer un balance de las ventajas y desventajas de cada una. Al final del proyecto, se debe tener un estudio exhaustivo del estado del arte, con pruebas y resultados reales.

### **4.2.2 SEGUNDO OBJETIVO**

El segundo objetivo es el de obtener un proceso que utilice técnicas de *Machine Learning* y *NLP* para caracterizar textos de tal manera que puedan relacionarse aquellos que tratan los mismos temas. Es decir, dos textos distintos que traten un mismo tema deben generar un resultado suficientemente parecido como para que puedan ser relacionados. Esto debe aplicarse a cualquier texto, incluidos aquellos que forman, por ejemplo, un test de evaluación con preguntas y varias opciones de respuesta. Esto involucra tanto programas de procesamiento de lenguaje natural como de tratamiento de texto. La solución final debe ser una combinación de las que ya existen, una adaptación de alguna de ellas o, en su defecto, una desarrollada desde cero.

### **4.2.3 TERCER OBJETIVO**

El tercer objetivo es el de satisfacer las especificaciones del proyecto de la empresa *The Wise Seeker*. Este proyecto busca facilitar el mejor curso de formación al usuario, tras haber realizado sin éxito una evaluación concreta. El servicio debe interpretar el título, la descripción y las materias de los cursos para casar el más adecuado con la evaluación que ha llevado a cabo el usuario. Debe darse una solución a esta funcionalidad, que se integre e implemente correctamente dentro de la plataforma de *The Wise Seeker*.

## **4.3 METODOLOGÍA**

El trabajo se dividirá en tres fases principales: Análisis, Selección, y Desarrollo

### **4.3.1 FASE DE ANÁLISIS**

En primer lugar, se busca llevar a cabo un análisis de las distintas alternativas que se ofrecen en el mercado a la hora de analizar textos de cara a poder ser comparados con otros. Deben estudiarse todas las opciones disponibles con el fin de filtrar las más adecuadas. Para ello, es necesario realizar un balance de las ventajas y desventajas de cada una y ver de qué manera podrían reutilizarse en el proyecto. Esto supone un trabajo de búsqueda e investigación.

### **4.3.2 FASE DE SELECCIÓN**

En esta segunda fase, se realizarán pruebas sobre las diferentes alternativas, con el fin de obtener las que más se adaptan al objetivo del proyecto. Esto supondrá poner a prueba, de manera práctica, las diferentes técnicas para dar con las más viables. Al final de esta etapa, se debe tener una selección de las tecnologías que se van a utilizar para el desarrollo del proyecto.

### **4.3.3 FASE DE DESARROLLO**

En esta última fase, se desarrollará la solución que cumpla con el objetivo del proyecto. Se diseñará un programa que adapte y combine las soluciones filtradas en la fase de selección.

El objetivo es optimizar la solución para que cumpla con los requisitos de la mejor manera posible. También deberá comprobarse la satisfacción del requisito de *The Wise Seeker*, haciendo las modificaciones del programa necesarias para su implementación.

#### ***4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA***

A continuación, se explica la planificación del proyecto y la estimación económica.

##### **4.4.1 PLANIFICACIÓN**

En la *Figura 8* se muestra la distribución temporal de las etapas mencionadas anteriormente. Como se puede ver, la etapa de análisis del estado del arte tiene un gran peso en la planificación, pues realizarla correctamente supone un punto de partida clave en el proyecto. La etapa de selección de tecnologías se solapa con el último mes de la etapa anterior, puesto que las pruebas en la segunda etapa pueden llevar a nuevos análisis en la primera. El primer tercio de la fase de desarrollo coincide con la segunda etapa ya que, al desarrollar el proyecto, se ponen en práctica las tecnologías seleccionadas. Esto puede desvelar nuevas desventajas de las herramientas que se están utilizando, lo que llevaría a hacer un cambio en la selección de tecnologías.

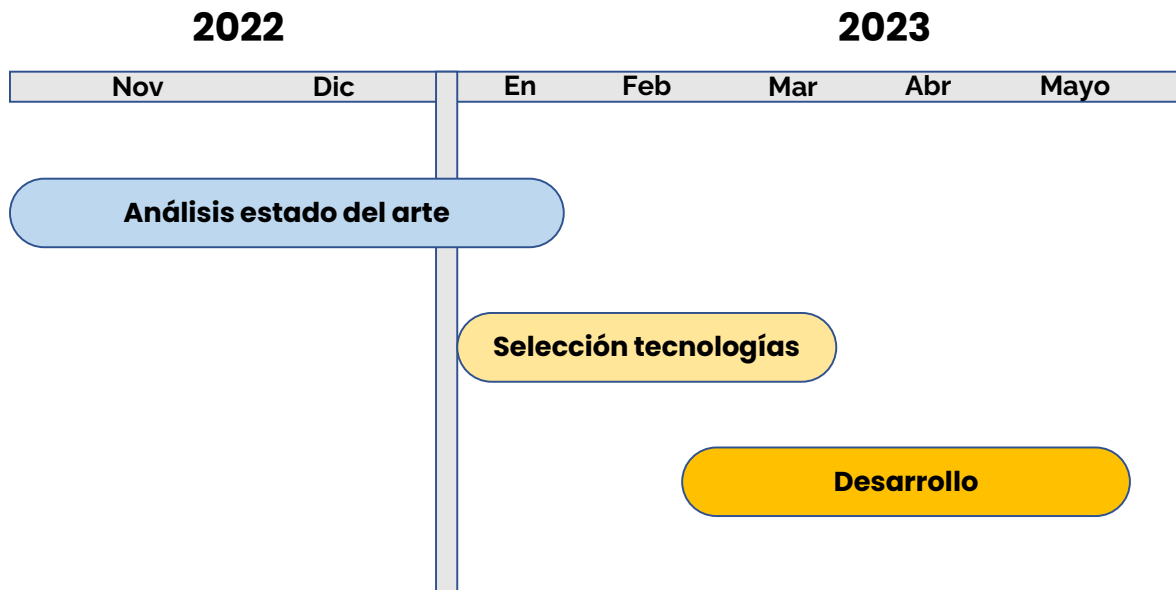


Figura 8. Planificación del proyecto

#### 4.4.2 ESTIMACIÓN ECONÓMICA

Puesto que este proyecto es completamente digital, no requiere de la compra de materiales físicos. El coste de su desarrollo viene exclusivamente de los servicios *API* que se quieran utilizar en la segunda y tercera etapa. En la fase de selección de tecnologías, es conveniente invertir en hacer pruebas con varias *API* para obtener una mejor imagen de las capacidades de las herramientas existentes. El precio de estos servicios depende directamente de la empresa que los proporciona. Algunos como *Google Cloud* ofrecen versiones gratuitas que son suficientemente capaces para este caso. Los que no son gratuitos cobran por número de tokens procesados, siendo cada palabra un token, aproximadamente. De cualquier modo, las *API* están orientadas al procesamiento de grandes cantidades de tokens, así que el coste del proyecto nunca será elevado debido a su menor magnitud. Haciendo una media del precio por token, se estima que el coste total del procesamiento de todos los textos necesarios para llevar a cabo el proyecto no supere los 50€. Este presupuesto es suficiente para pasar unos 15 000 textos de una media de 200 palabras por 4 *API* diferentes.

## Capítulo 5. DESARROLLO DEL PROYECTO

En este capítulo, se muestra el desarrollo de cada una de las fases descritas en la *Sección 4.3*. Sin embargo, antes es necesario definir de manera concreta los requisitos del programa que se busca obtener.

### 5.1 *ESPECIFICACIÓN DE REQUISITOS*

Los siguientes requisitos describen la funcionalidad y las capacidades que debe tener el sistema desarrollado.

Con el fin de aclarar las explicaciones, se definen dos tipos de textos dentro del sistema:

- Texto de entrada: Es el texto único que se introduce en el sistema, del que se desea obtener textos con temas similares. Este texto se introduce desde fuera y no queda almacenado en el sistema.
- Textos objetivo: Son los que se deben comparar con el texto de entrada, para determinar si comparten un mismo tema. Son los posibles textos de salida. Puede haber decenas de miles de textos objetivo. Estos están almacenados en el sistema.

#### 5.1.1 REQUISITOS FUNCIONALES

Estos requisitos describen la funcionalidad que debe cumplir el programa:

- El sistema debe recibir un texto de entrada, así como una lista de textos objetivo que se desean comparar con este. Debe caracterizar el texto de entrada y compararlo con los textos objetivo para devolver los  $n$  textos de la lista cuyo tema sea más similar al de entrada.
- El sistema debe permitir la selección del idioma de los textos objetivo a recomendar.
- Todo el sistema debe ser ejecutable mediante funciones de Python.

- Si no existen textos objetivo que traten el tema del texto de entrada, el sistema no debe devolver ningún elemento.

### **5.1.2 REQUISITOS DE ESTRUCTURA**

Estos requisitos describen la estructura de bloques que debe seguir el sistema.

Para lograr la funcionalidad, el sistema debe estar compuesto por tres bloques principales: bloque de caracterización, bloque de comparación y bloque de selección.

El bloque de caracterización es el encargado de generar un elemento que identifique al tema del texto de entrada. Este elemento puede ser de cualquier tipo o forma, pero debe representar correctamente el tema que trata el texto introducido. Por ejemplo, podría tratarse de una etiqueta que contenga palabras clave sobre el tema del texto. A partir de este bloque, el sistema trabaja con el resultado de la caracterización del texto en vez de con el texto al completo. A su vez, todos los textos objetivo deben haber pasado previamente por este bloque, por lo que sus caracterizaciones deben estar ya almacenadas.

El bloque de comparación se encarga de comparar la caracterización del texto de entrada con las caracterizaciones almacenadas de los textos objetivo. Mediante este proceso, se obtiene qué textos objetivo tienen una caracterización más similar a la del texto de entrada y, por tanto, cuáles tratan un tema más parecido. El cálculo de esta similitud puede hacerse utilizando la distancia de coseno con *embeddings* (ver Capítulo 2. ), o de cualquier otra manera que compare las caracterizaciones. El resultado del bloque comparador debe ser una lista de valores numéricos que representen cómo de parecido es el tema que trata cada texto objetivo con el tema del texto de entrada.

Finalmente, el bloque de selección recibe los resultados de las comparaciones de cada texto objetivo con el texto de entrada, y debe decidir qué textos objetivo devolver a la salida del sistema.

En la *Figura 9* se puede ver una representación de la estructura que debe seguir el sistema.

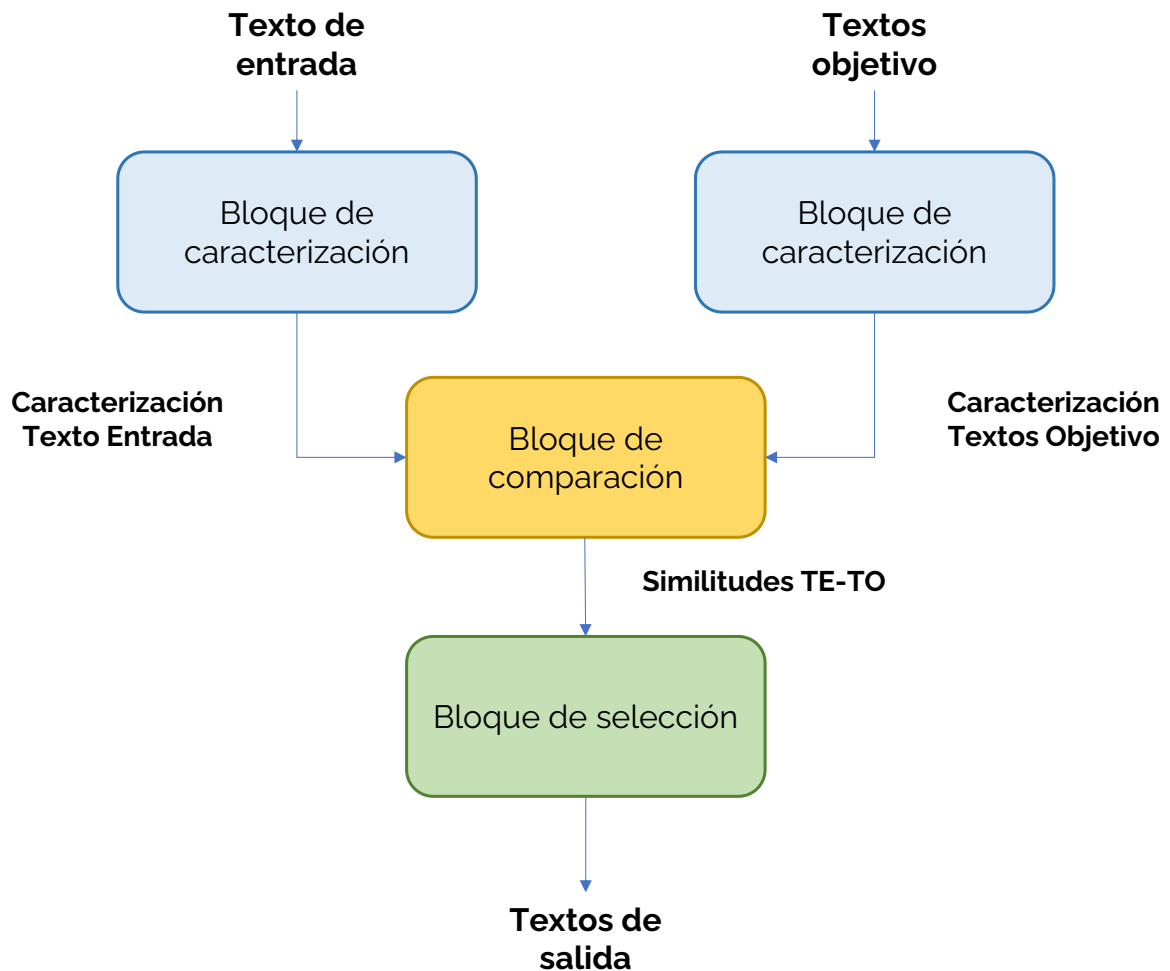


Figura 9. Diagrama de bloques del sistema

### 5.1.3 REQUISITOS NO FUNCIONALES

Estos requisitos describen las capacidades y limitaciones que debe tener el programa:

- El sistema no debe requerir de entrenamiento. Debe funcionar desde su primera implementación.



- El sistema debe mostrar alta flexibilidad en cuanto al formato y contenido de los textos. Debe ser capaz de comparar textos que representen diferentes elementos como: cursos online, películas, libros, exámenes tipo test, etc.
- El sistema debe ser capaz de trabajar con decenas de miles de textos objetivo. La cantidad de textos no debe ser un factor limitante.
- El sistema debe devolver los  $n$  textos objetivo con tema más similar al de entrada en un tiempo razonable, siendo este de menos de 5 segundos.
- El sistema debe dar respuestas con, por lo menos, un 90% de precisión. Es decir, los textos devueltos deben mostrar un tema similar al de entrada en un 90% de las veces.
- El sistema debe permitir almacenar las caracterizaciones de los textos para no tener que llamar a las *API* en cada ejecución.
- El único factor limitante debe ser la longitud de los textos. Esta no puede ser demasiado corta, pues los textos tendrían poca información. Tampoco puede ser demasiado larga, pues las *API* de *NLP* tienen un límite de palabras por llamada. Así pues, la longitud mínima debe ser de 3 palabras, y la máxima de 1000 palabras.
- Además, el sistema debe poder comparar textos de distintas longitudes, siempre y cuando cumplan con la mínima y la máxima.
- El sistema debe ser capaz de trabajar en varios idiomas como el español, el inglés, el francés, o el alemán, sin que la precisión se vea afectada.

## **5.2 FASE DE ANÁLISIS DEL ESTADO DEL ARTE**

En el *Capítulo 3. Estado de la cuestión*, se desarrolla el estado del arte actual del procesamiento del lenguaje natural. En esta fase del desarrollo, se pretende enfocar más el análisis sobre el objetivo concreto del proyecto. Se trata de analizar qué tecnologías pueden ser útiles para cumplir con las dos funcionalidades que se buscan: la caracterización de los textos y la comparación entre los mismos. Así pues, se procede a analizar cada una de las

opciones descritas en el estado de la cuestión, para determinar si conviene hacer pruebas sobre ellas en la siguiente fase.

### **5.2.1 LDA**

El *LDA* es uno de los algoritmos más usados dentro de la caracterización de textos pues permite obtener una descripción de un documento en función de los temas detectados. Esto puede parecer muy adecuado para el proyecto, pero plantea dos problemas principales.

Por un lado, este algoritmo es uno de los más antiguos dentro del mundo de *NLP*, pues fue presentado en enero del año 2003 [2]. Esto no es necesariamente algo negativo, pero este proyecto busca emplear las últimas tecnologías dentro de un ámbito que muestra una rápida evolución a lo largo de los años.

Por otro lado, a priori, la naturaleza del algoritmo *LDA* es no supervisada, es decir, no requiere de entrenamiento. Sin embargo, para una implementación completa del algoritmo dentro de un sistema concreto, sí que es necesario entrenar el modelo para obtener resultados adecuados. [3]

Por estas razones, queda descartada la opción de utilizar el *LDA* en este proyecto.

### **5.2.2 SPACY**

*Spacy* [4] es otra de las herramientas gratuitas de análisis de texto en *Python* mediante *NLP*. A través de su librería, ofrece multitud de opciones a la hora de extraer información de un texto. Concretamente, posee dos ventajas que pueden resultar clave para el proyecto.

Por un lado, *Spacy* ofrece la funcionalidad de generar *embeddings* de palabras o frases. Esto permite comparar diferentes textos para obtener el grado de similitud que existe entre ellos. Además, la implementación de esta funcionalidad es sencilla gracias a las funciones de la librería de *Spacy*.

Por otro lado, *Spacy* posee modelos de lenguaje ya entrenados, por lo que no sería necesaria una fase de entrenamiento. Esto cuadra perfectamente con lo que se busca en el proyecto.

Gracias a su funcionalidad de comparar textos mediante *embeddings* y a sus modelos de lenguaje ya entrenados, es conveniente realizar pruebas prácticas con *Spacy* para finalmente decidir si se debe usar en el proyecto.

### 5.2.3 GOOGLE CLOUD NATURAL LANGUAGE

*Google* ofrece una *API* de procesamiento de lenguaje natural que merece la pena estudiar para el proyecto. Debido al prestigio de la empresa, es esperable que esta tecnología sea una de las más actualizadas y utilizadas en el sector. En cuanto a este proyecto, muestra dos importantes puntos a favor.

La *API* ofrece varias funcionalidades a la hora de analizar textos, pero hay una de ellas que es atractiva de cara a la caracterización de documentos. Esta funcionalidad es la de clasificación, que categoriza el texto en una de las 700 categorías ya predefinidas por *Google*. En la *Figura 10* se muestra un ejemplo.

### Try the API

This is a course about databases and SQL queries. It aims to teach how to manage relational databases by using SQL syntax and tools like MySQLServer. You will learn how to set up a database with primary keys and foreign keys, while managing tables with multiple columns with different types of data.

[See supported languages](#)

Entities	Sentiment	Syntax	Categories
			<p><b>/Computers &amp; Electronics/Enterprise Technology/Data Management</b> Confidence: 0.85</p> <p><b>/Science/Computer Science</b> Confidence: 0.56</p> <p><b>/Business &amp; Industrial</b> Confidence: 0.73</p> <p><a href="#">See a complete list of content categories.</a></p>

Figura 10. Ejemplo de clasificación de textos con Google. [5]

Como se puede observar, se ha introducido el texto que representa la información de un curso online sobre bases de datos y *SQL*. El resultado de la clasificación es la categoría “*/Computers & Electronics/Enterprise Technology/Data Management*” con una confianza del 85%, además de otras dos categorías, pero con menos confianza. Este sencillo ejemplo demuestra que se podría usar esta *API* para caracterizar los textos, asignándoles a cada uno una categoría.

Además, este proceso no requiere de entrenamiento, pues se emplean los modelos de lenguaje ya entrenados por *Google*. Por todo esto, se realizarán pruebas con esta herramienta para comprobar su verdadero potencial.

#### 5.2.4 AMAZON COMPREHEND Y AZURE COGNITIVE SERVICE

Estas dos alternativas de *Amazon* [6] y de *Microsoft* [6] son realmente potentes, pero al analizarlas resultan no ser tan adecuadas para este proyecto.

Por su parte, *Amazon Comprehend* ofrece un servicio más técnico y personalizable, que no es lo que se busca en este caso. Requiere de fases de entrenamiento y de un alto nivel de configuración, con el fin de cubrir necesidades más complejas que las de este trabajo.

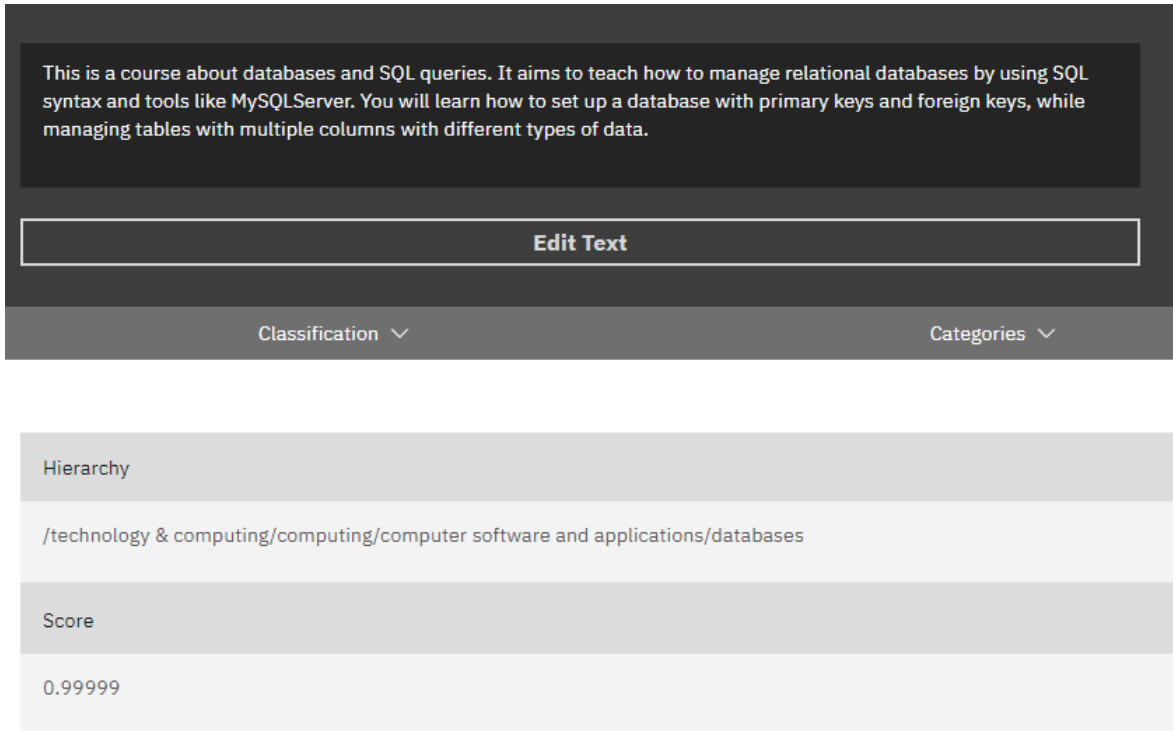
Por otra parte, *Azure Cognitive Service* posee una *API* similar a la de *Google*, pero enfocada en otras funcionalidades. No tiene la capacidad de categorizar o caracterizar textos, sino que se centra en, entre otras cosas, la generación de resúmenes de grandes documentos o el análisis de sentimientos y opiniones.

Por estas razones, estas dos tecnologías quedan descartadas.

#### 5.2.5 IBM WATSON

Otra de las opciones de las grandes empresas es la herramienta *Watson*, de *IBM* [7]. *Watson* lleva muchos años dentro del mundo del *NLP*, por lo que se ha forjado un gran prestigio. Analizando lo que ofrece, se observan dos grandes puntos a favor de esta herramienta.

Por un lado, muestra un funcionamiento y unas capacidades muy similares a las de la *API* de *Google*, lo que es algo positivo. Esto quiere decir que también otorga la posibilidad de clasificar documentos en categorías ya definidas, lo que se convierte en una manera sencilla y eficaz de caracterizar textos para este trabajo. En la *Figura 11* se muestra un ejemplo.



The screenshot shows a dark-themed interface for text classification. At the top, a text box contains the following text: "This is a course about databases and SQL queries. It aims to teach how to manage relational databases by using SQL syntax and tools like MySQLServer. You will learn how to set up a database with primary keys and foreign keys, while managing tables with multiple columns with different types of data." Below the text box is a white-bordered button labeled "Edit Text". At the bottom of the interface, there are two dropdown menus: "Classification" and "Categories".

Hierarchy
/technology & computing/computing/computer software and applications/databases

Score
0.99999

Figura 11. Ejemplo de clasificación de textos con IBM Watson [8]

En este ejemplo se ha introducido exactamente el mismo texto que en el de Google. El resultado es una única categoría “*technology & computing/computing/computer software and applications/databases*” con una confianza de prácticamente el 100%. Este resultado es muy bueno y, a priori, mejor que el de la API de Google, pues tiene mayor confianza y no devuelve otras categorías incorrectas.

Por otro lado, al igual que Google, emplea modelos ya entrenados, así que no requiere de ningún tipo de entrenamiento.

Por tanto, sin ninguna duda, es una alternativa que hay que probar.

## 5.2.6 OPENAI

Como se puede ver en la *Sección 4.4.1* (Planificación), este proyecto comenzó en noviembre de 2022 con un análisis del estado de la cuestión. En ese momento, también se hizo un estudio sobre las tecnologías que ofrecía la empresa *OpenAI* pues, a pesar de no ser de tanta importancia como las otras, merecía la pena ver si podía ofrecer algo. El resultado de este estudio no fue positivo, ya que las herramientas de análisis de textos de esta empresa no funcionaban bien. En comparación con las otras alternativas, los servicios de *OpenAI* eran poco precisos, más caros y más lentos [9], y se centraban en la generación de *embeddings*. *Spacy*, por ejemplo, ofrecía una funcionalidad similar con resultados muy superiores. Por todo esto, la opción de emplear *OpenAI* en el proyecto quedó descartada rápidamente.

Sin embargo, como ya se sabe, a principios de 2023 surgió una tecnología abierta al público que causó grandes sensaciones: *ChatGPT*. Esta herramienta, perteneciente a *OpenAI*, supuso el principio de una revolución dentro del mundo del *NLP*. Además, la misma empresa desarrolló simultáneamente nuevas versiones de su modelo generador de *embeddings* [10], que ahora implementaba la nueva arquitectura *GPT*. Por último, *OpenAI* también lanzó una *API* que habilitaba la conexión con *ChatGPT* a través de peticiones *HTTP*, permitiendo así su implementación dentro de aplicaciones [11].

Por todo esto, hubo que replantear la idea de emplear los servicios de esta empresa, pues se habían visto drásticamente mejorados. Así pues, ahora *OpenAI* aportaba mucho a la cuestión del proyecto gracias a sus dos funcionalidades. Por un lado, el nuevo generador de *embeddings* podía competir con *Spacy*, siendo una opción interesante a la hora de comparar textos. Por otro lado, la *API* de *ChatGPT* abría un gran abanico de posibilidades de cara a la caracterización de textos con capacidades como la extracción de palabras clave. Por tanto, las herramientas de *OpenAI* pasan a ser fundamentales para este proyecto y deben ser probadas en la siguiente fase.

### 5.2.7 CONCLUSIONES DEL ANÁLISIS DEL ESTADO DE LA CUESTIÓN

Tras haber analizado todas las opciones que podrían servir como base para este proyecto, se decide que se va a dar continuidad al estudio de las siguientes tecnologías:

- *Spacy*, por su generación de *embeddings* que puede servir para comparar textos.
- *Google Cloud*, por su clasificación de textos, que puede servir para caracterizarlos.
- *IBM Watson*, por su clasificación de textos, que puede servir para caracterizarlos.
- *OpenAI*, por su generación de *embeddings*, y por la nueva *API* de *ChatGPT*.

Entonces, se puede proceder a la siguiente fase, en la que se deben hacer pruebas prácticas para determinar finalmente qué tecnologías se van a usar para cumplir la funcionalidad del proyecto.

....

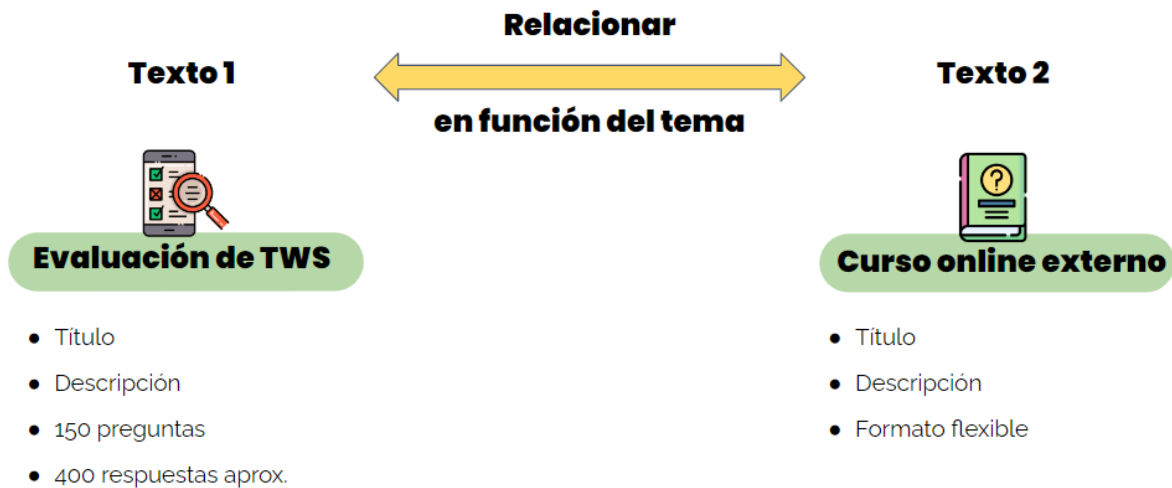
## 5.3 FASE DE SELECCIÓN DE TECNOLOGÍAS

### 5.3.1 CONTEXTO DE LAS PRUEBAS

En esta fase, la idea es hacer pruebas prácticas con las cuatro tecnologías para acabar teniendo un diseño que cumpla los requisitos del proyecto. Para esto, se va a trabajar en todo momento con un contexto concreto, que coincide con el entorno de la empresa colaboradora *The Wise Seeker*. Este contexto busca hacer una recomendación de cursos online a un usuario que acaba de realizar un examen sobre un tema concreto. Entonces, se busca asociar un elemento que represente un examen con otros elementos que representen cursos online. Los cursos online se obtienen de diferentes empresas proveedoras y son cursos reales. Este



ejemplo de aplicación pone a prueba el proyecto, pues los elementos a asociar son muy distintos, como se puede ver en la *Figura 12*.



*Figura 12. Elementos a asociar*

Siguiendo la estructura descrita en la *Sección 5.1.2*, el texto de entrada sería la evaluación de *The Wise Seeker*, mientras que los textos objetivo serían los cursos online externos.

Si los cursos recomendados llegan a casar correctamente con el tema del examen que el usuario ha realizado, demostraría que el modelo es capaz de asociar elementos de formato muy flexible y, por tanto, iría por buen camino. Por tanto, todas las pruebas que se realicen deben ir enfocadas a esta funcionalidad.

### 5.3.2 PRUEBAS DE CARACTERIZACIÓN

Una vez definido el contexto, se pueden empezar las pruebas para ver qué tecnologías pueden dar vida a los bloques de caracterización y de comparación. Como el bloque de comparación recibe como entrada las caracterizaciones de los textos, se debe empezar por el bloque de caracterización. Para este bloque, se pueden emplear dos posibles tecnologías: *Google Cloud* e *IBM Watson*, teniendo que seleccionar una de ellas.

### 5.3.2.1 Caracterización con Google Cloud

Como se menciona en la *Sección 5.2.3*, *Google Cloud* ofrece la caracterización de un texto mediante su funcionalidad de clasificación en una o varias de las 700 categorías predefinidas por *Google* [12]. Las categorías extraídas de un texto pueden servir para definir su tema principal, por lo que se deben hacer pruebas para comprobarlo. Entonces, es necesario comprobar que el resultado de la clasificación define correctamente tanto el tema del texto de entrada (evaluación de *TWS*) como el tema de los textos objetivos (cursos online).

Empezando por los posibles textos de entrada, en las siguientes tablas se muestran los resultados de las clasificaciones de unas evaluaciones de *TWS* que tratan temas tecnológicos. La primera columna muestra el título de la evaluación, la segunda columna muestra la categoría obtenida tras introducir en *Google Cloud* toda la información de la evaluación, y la tercera columna muestra la confianza de cada clasificación. La cabecera de cada tabla indica el tema general que engloba a las evaluaciones.

<b>CIBERSEGURIDAD</b>		
Ciberseguridad	/Computers & Electronics/Networking	97%
Seguridad de la información	/Computers & Electronics/Computer Security	93%
Seguridad Electrónica	/Law & Government/Public Safety/Security Products & Services	92%
	/Arts & Entertainment/Visual Art & Design/Photographic & Digital Arts	86%
	/Computers & Electronics/Consumer Electronics/Camera & Photo Equipment	72%
	/Hobbies & Leisure	66%

<b>DATOS</b>		
Estadística	/Science/Mathematics/Statistics	99%
Analítica de datos	/Computers & Electronics/Enterprise Technology/Data Management	91%

	/Business & Industrial	87%
Bases de datos	/Computers & Electronics/Enterprise Technology/Data Management	98%
	/Business & Industrial	92%
Machine Learning	/Science/Computer Science	98%
Big Data	/Computers & Electronics/Enterprise Technology/Data Management	93%
	/Business & Industrial	77%

SISTEMAS		
Redes	/Computers & Electronics/Networking	99%
Routing	/Computers & Electronics/Networking/Data Formats & Protocols	98%
Arquitectura Software	/Computers & Electronics/Software	75%
	/Science	59%
Cloud	/Internet & Telecom/Web Services	73%
	/Business & Industrial	64%
	/Computers & Electronics/Networking	50%
Internet of things	/Computers & Electronics/Networking	75%
	/Internet & Telecom	70%
Ingeniería de software	/Computers & Electronics/Software	76%
Voz sobre IP	/Internet & Telecom/Email & Messaging/Voice & Video Chat	75%
Digitalización	/Computers & Electronics/Software/Business & Productivity Software	57%
Blockchain	/Finance/Investing/Currencies & Foreign Exchange	99%
Medios de Pago	/Business & Industrial/Business Services/E-Commerce Services	95%

	/Computers & Electronics/Enterprise Technology	86%
	/Finance	65%
CRM ERP	/Business & Industrial	99%
	/Computers & Electronics/Enterprise Technology	99%
	/Computers & Electronics/Software/Business & Productivity Software	50%

Tabla 1. Clasificación con Google Cloud de los textos de Entrada

Tras analizar los resultados, se observa que las categorías coinciden correctamente con la evaluación en un 70% de los casos. Las evaluaciones de “Ciberseguridad”, “Seguridad Electrónica”, “Machine Learning”, “Cloud”, e “Internet of things” no muestran un resultado tan satisfactorio. Esto es porque la categoría obtenida no representa el tema de la evaluación, o porque hay otras categorías no obtenidas que lo representan con mucha más precisión.

De todos modos, esto no es necesariamente negativo, pues la caracterización en sí misma no es tan relevante. Lo importante es que las caracterizaciones de dos textos de tema similar sean muy parecidas. Es decir, tiene mayor peso la diferencia entre caracterizaciones, que cada caracterización individual.

A continuación, se muestran los resultados de las mismas pruebas, pero esta vez trabajando con los textos objetivo, que representan cursos online.

Bases de datos con SQL Server	/Computers & Electronics/Enterprise Technology/Data Management	97%
	/Business & Industrial	93%
Ciberseguridad. Protege tu información de Cibercriminales.	/Computers & Electronics/Computer Security	80%
	/Jobs & Education/Education	60%

Angular PRO desde cero: El curso definitivo (Angular 8+)	<b>/Science</b>	94%
Aprende a programar jugando con Scratch	<b>/Jobs &amp; Education/Education</b>	86%

Tabla 2. Clasificación de textos objetivo

La Tabla 2 muestra varios casos interesantes de entre 900 pruebas.

En la primera fila, el curso online de Bases de datos se clasifica correctamente. Además, la categoría obtenida coincide con la categoría del texto de entrada que también trataba sobre bases de datos, por lo que el sistema estaría funcionando en este caso.

En la segunda fila, el curso sobre ciberseguridad devuelve una categoría que es adecuada a su tema. Esta también coincide con la categoría obtenida con el texto de entrada sobre ciberseguridad. No obstante, la confianza del resultado es menor (80%), y también aparece una categoría que no describe el tema del curso (“/Jobs & Education/Education”).

En la tercera fila, se observa un resultado negativo. El modelo de Google no es capaz de clasificar correctamente el curso, por lo que le asigna una categoría completamente general que no describe nada concreto (“/Science”).

En el último caso, el curso de programación con Scratch genera una categoría que no corresponde con su tema principal: “Jobs & Education/Education”. El modelo de Google devuelve esta categoría porque la descripción de este curso contiene muchas expresiones que pertenecen al tema del aprendizaje como “aprende a” o “con este curso aprenderás”. En este caso, la categoría obtenida define el concepto de curso, pero no el concepto del tema principal del curso. Este problema surge con frecuencia, pues casi un 20% de los cursos reciben esta caracterización.

Por otro lado, en algunos casos la API de Google Cloud no es capaz de clasificar el texto objetivo porque no llega a la longitud mínima. Como se puede ver en la Figura 13, una descripción breve del curso online no genera ninguna categoría debido a la escasa

información. Esto puede ser un problema, pues hay algunos textos objetivo que resumen muy brevemente su contenido.

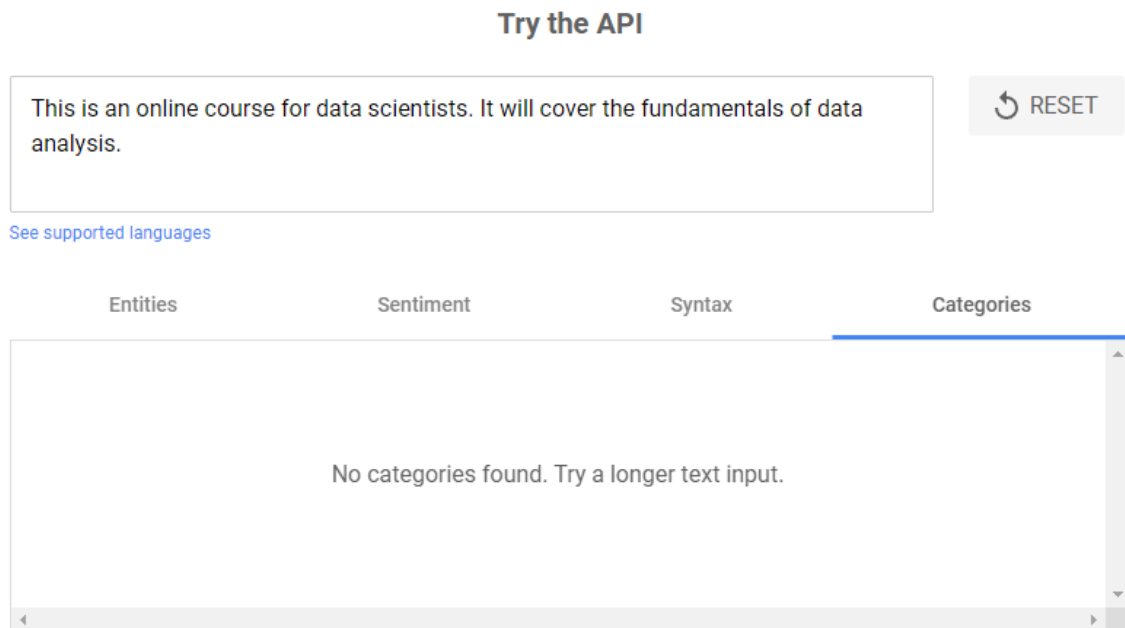


Figura 13. Problema de longitud de texto mínima con Google

En definitiva, la tecnología de *Google* muestra los siguientes problemas a la hora de caracterizar los textos de entrada y los textos objetivo:

- La categoría obtenida no coincide con precisión con el tema del texto.
- La confianza puede ser baja.
- La categoría obtenida es demasiado general y poco específica.
- La categoría obtenida describe el concepto de curso en vez del tema del curso.
- Un texto corto no genera ninguna clasificación.

### 5.3.2.2 Caracterización con IBM Watson

Como se menciona en la *Sección 5.2.5*, *IBM Watson* también posee la capacidad de clasificar textos en categorías predefinidas. De la misma manera que *Google*, esta tecnología puede servir para dar funcionalidad al bloque de caracterización. Así pues, es necesario hacer las mismas pruebas con esta alternativa.

Empezando con los posibles textos de entrada, en las siguientes tablas se muestran las clasificaciones obtenidas para las evaluaciones de ámbito tecnológico de *The Wise Seeker*. Al igual que anteriormente, la primera columna indica el título de la evaluación, la segunda muestra la categoría resultante de la clasificación, y la tercera la confianza.

CIBERSEGURIDAD		
Ciberseguridad	/technology & computing/computing/computer networking	85%
	/technology & computing/computing/information and network security	81%
Seguridad de la información	/technology & computing/computing/information and network security	97%
Seguridad Electrónica	/technology & computing/computing/computer software and applications	84%
	/technology & computing/computing/information and network security	83%

DATOS		
Estadística	/technology & computing/computing/computer software and applications/digital audio	79%
Analítica de datos	/technology & computing/computing/data storage and warehousing	90%
	/business and finance/business/business i.t.	84%
Bases de datos	/technology & computing/computing/computer software and applications/databases	100%

Machine Learning	/technology & computing/artificial intelligence	96%
Big Data	/technology & computing/computing/computer software and applications/databases	99%

SISTEMAS		
Redes	/technology & computing/computing/computer networking	99%
Routing	/technology & computing/computing/computer networking	99%
Arquitectura Software	/technology & computing/computing/computer software and applications/databases	99%
	/technology & computing/computing/computer software and applications/operating systems	99%
	/technology & computing/computing/internet/web hosting	89%
Cloud	/technology & computing/computing/computer software and applications/databases	89%
	/technology & computing/computing/computer software and applications/operating systems	72%
	/technology & computing/computing/data storage and warehousing	65%
Internet of things	/technology & computing/computing/internet/internet of things	98%
	/technology & computing/computing/computer software and applications	93%
Ingeniería de software	/technology & computing/computing/computer software and applications	95%
	/technology & computing/computing/computer software and applications/operating systems	88%
Voz sobre IP	/technology & computing/computing/internet/email	73%
Digitalización	/technology & computing/computing/computer software and applications	96%
	/video gaming/mobile games	86%
	/technology & computing/computing/computer software and applications/operating systems	86%
Blockchain	/technology & computing/computing/information and network security	83%



Medios de Pago	<b>/personal finance/personal debt/credit cards</b>	99%
CRM ERP	<b>/technology &amp; computing/computing/computer software and applications</b>	79%
	<b>/business and finance/business/business i.t.</b>	74%
	<b>/business and finance/industries/information services industry</b>	74%

Tabla 3. Clasificación con IBM de los textos de entrada

Analizando los resultados, se observa que hay cinco evaluaciones clasificadas incorrectamente: “Estadística”, “Arquitectura Software”, “Voz sobre IP”, “Blockchain” y “Medios de Pago”. Esto supone un 75% de precisión. De nuevo, estos fallos se dan porque la categoría obtenida no representa el contenido de la evaluación, o porque hay otras categorías que lo representan con más precisión.

Atendiendo ahora a los textos objetivo, que representan cursos online, en la *Tabla 4* se muestran los resultados de las clasificaciones de algunos de los 900 textos objetivo. Estos son exactamente los mismos que los de *Google Cloud*.

Bases de datos con SQL Server	<b>/technology &amp; computing/computing/computer software and applications/databases</b>	99%
Ciberseguridad. Protege tu información de Ciberdelincuentes.	<b>/Education</b>	90%
	<b>/Education/online education</b>	75%
Angular PRO desde cero: El curso definitivo (Angular 8+)	<b>/technology &amp; computing/computing/internet/web development</b>	94%
Aprende a programar jugando con Scratch	<b>/Education</b>	99%

Tabla 4. Clasificación con IBM de textos objetivo

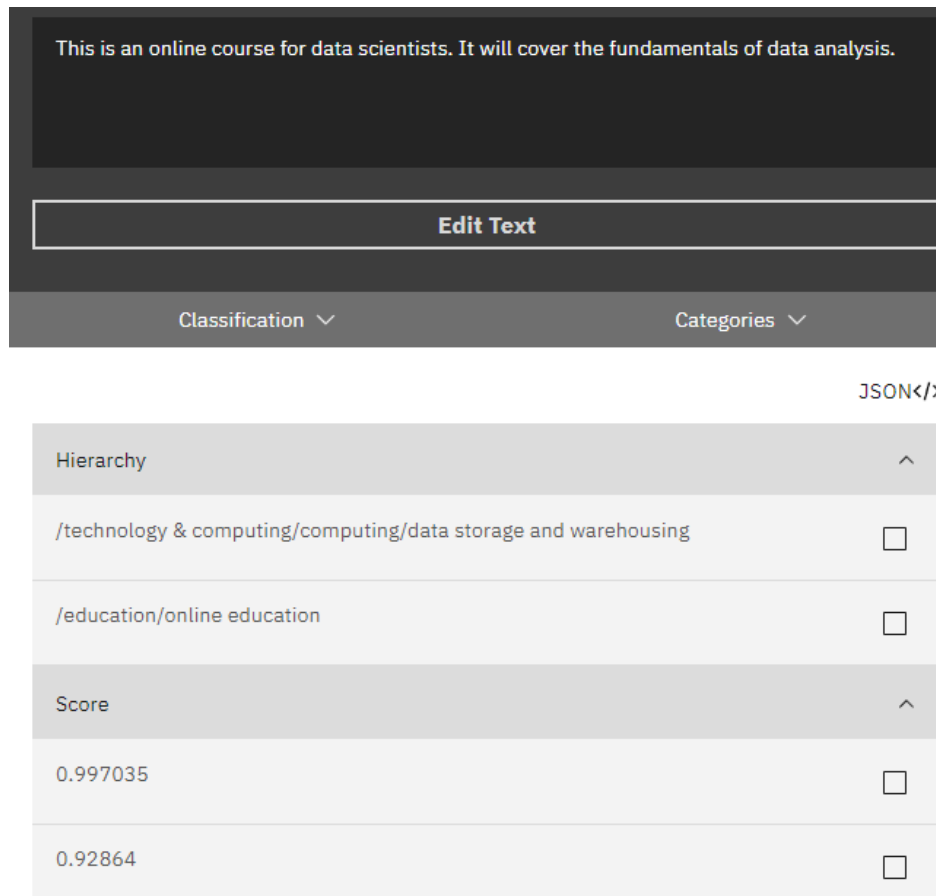
En el primer caso, la clasificación es correcta y con mucha confianza. El curso trata sobre bases de datos, y la categoría resultante es exactamente esa.

En el segundo caso, se observa que el modelo de IBM devuelve la categoría que corresponde con el concepto literal de curso y aprendizaje. En vez de caracterizar el tema del curso, está caracterizando lo que es: un curso de aprendizaje. Por tanto, el resultado es negativo.

El tercer caso es una clasificación acertada, pues el curso trata sobre Angular y la categoría obtenida es la de desarrollo web. Además, la confianza es bastante alta (94%).

En el cuarto texto se observa el mismo problema que en el segundo, por lo que tampoco es correcto. Este problema surge en un 20% de los casos aproximadamente.

Finalmente, queda comprobar qué hace el modelo de IBM cuando se le introduce un texto corto. Como se puede ver en la *Figura 14*, el resultado es satisfactorio. No solamente es capaz de clasificar el texto sino que, además, la primera categoría obtenida es correcta.



This is an online course for data scientists. It will cover the fundamentals of data analysis.

Edit Text

Classification ▾ Categories ▾

JSON</>

Hierarchy	
/technology & computing/computing/data storage and warehousing	<input type="checkbox"/>
/education/online education	<input type="checkbox"/>
Score	
0.997035	<input type="checkbox"/>
0.92864	<input type="checkbox"/>

Figura 14. Clasificación con IBM de un texto corto

### 5.3.2.3 Comparativa de Google Cloud con IBM Watson

Habiendo analizado los resultados de las caracterizaciones de los textos de entrada y los textos objetivo, se llegan a las siguientes conclusiones sobre *Google Cloud* e *IBM Watson*.

- Ambas alternativas muestran resultados con una precisión del 70%-75%.
- Ambos modelos clasifican el concepto de aprendizaje en vez del tema del curso en un 15-20% de los textos objetivo.
- En los resultados correctos, *IBM Watson* acierta con mayor confianza.
- *Google Cloud* requiere un texto más largo que *IBM Watson*.

El problema de Google Cloud de la longitud mínima de texto es realmente limitante, pues el modelo debe dar resultado para todos los textos objetivo. Puesto que ambas tecnologías rinden de manera similar en el resto de aspectos, este factor es decisivo.

Por tanto, se selecciona la tecnología de *IBM Watson* para el bloque de caracterización, y se descarta la de *Google Cloud*.

#### **5.3.2.4 Caracterización con ChatGPT**

Como se menciona en la *Sección 5.2.6*, *OpenAI* también ofrece una manera de caracterizar los textos mediante su tecnología *ChatGPT*. Esta potente herramienta puede recibir un texto de entrada y unas instrucciones sobre lo que tiene que devolver. Entonces, se le puede pedir que caracterice el texto a través de la selección de cinco palabras clave que describan el tema del texto.

Para trabajar con la API de *ChatGPT*, simplemente se introduce en la llamada el mensaje, y se recibe la respuesta de la inteligencia artificial. En este caso, el mensaje debe estar formado por: las instrucciones de análisis y el texto a analizar.

Las instrucciones de análisis deben ser lo suficientemente claras como para que *ChatGPT* devuelva exactamente cinco palabras que describan el tema del texto que acompaña el mensaje. Además, aquí surge la oportunidad de indicar qué representa el texto que se desea caracterizar. En el caso de los textos de entrada, se puede mencionar en el mensaje que representan un examen, mientras que, en los textos objetivo, se puede indicar que representan un curso online. De esta manera, *ChatGPT* entiende lo que describe cada texto, para así poder caracterizarlo con mayor precisión.

Empezando por los textos de entrada, la instrucción que se le puede dar a la inteligencia artificial es la siguiente:

*“Esta es la información de un test online. Escribe en una línea y en inglés las 5 palabras clave que describen el tema que trata el test.”*

Seguidamente, se incluye en el mensaje el texto de entrada. También es importante notar que se pide que las palabras devueltas sean siempre en inglés, lo que hace que no importe el idioma de los textos de entrada y los textos objetivo, siempre y cuando *ChatGPT* pueda comprenderlo.

En la *Tabla 5* se muestran las cinco palabras o conceptos clave que describen el tema de cada texto de entrada.

<b>CIBERSEGURIDAD</b>	
Ciberseguridad	<b>Cybersecurity, awareness, threats, protection techniques, encryption.</b>
Seguridad de la información	<b>Cybersecurity, Information Security, Data Protection, Threats, Protection Techniques.</b>
Seguridad Electrónica	<b>Electronic security, video surveillance, access control systems, alarm systems, technology.</b>

<b>DATOS</b>	
Estadística	<b>Statistics, Data Analysis, Statistical Models, Statistical Inference, Data Sources.</b>
Analítica de datos	<b>Data analytics, analytical models, business intelligence, web analytics, machine learning.</b>
Bases de datos	<b>Databases, Fundamental concepts, Relational model, Structured databases, Unstructured databases.</b>
Machine Learning	<b>Machine Learning, concepts, techniques, models, bots.</b>
Big Data	<b>Big Data, Systems architecture, Ingestion, Analytics, Data security.</b>

<b>SISTEMAS</b>	
Redes	networks, WAN/LAN, SDN technology, WLAN/WiFi, mobile networks
Routing	Routing, OSI model, IP addressing, TCP/IP protocols, Systems.
Arquitectura Software	Software architecture, systems architecture, operating systems, SOA architectures, microservices.
Cloud	Public cloud, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Serverless computing.
Internet of things	Internet of Things, connectivity, sensorization, security models, industry verticals.
Ingeniería de software	Software Engineering, Version Control, Object-Oriented Programming, Concurrency, Debugging.
Voz sobre IP	VoIP, SIP protocol, voice compression, gateways, systems.
Digitalización	Digitization, Digital transformation, Tools, Applications, Workplace.
Blockchain	Blockchain, P2P Networks, Cryptography, Digital Assets, Smart Contracts.
Medios de Pago	Payment methods, Fraud, Cybercrime, Regulations, Open Banking.
CRM ERP	CRM, ERP, systems, implementation, comparison.

*Tabla 5. Palabras clave de textos de entrada con ChatGPT*

Como se puede ver, todas las palabras clave obtenidas describen el tema de cada uno de los textos de entrada, por lo que se puede decir que la precisión es del 100%. Aunque no todas las palabras sean específicas del tema del texto, el conjunto de las cinco palabras sí que describe ese tema.

A continuación, se procede a hacer el mismo ejercicio con los textos objetivo. Como ahora los textos representan cursos online en vez de exámenes, es necesario cambiar la instrucción

que se le da a *ChatGPT* para que sepa interpretar el contenido. Así pues, la nueva instrucción pasaría a ser:

*“Esta es la información de un curso online. Escribe en una línea y en inglés las 5 palabras clave que describen el tema que trata el curso.”*

Solamente se ha cambiado la palabra “test” por la palabra “curso”. Con este simple cambio, el modelo *NLP* de *ChatGPT* ya puede entender el contenido del texto.

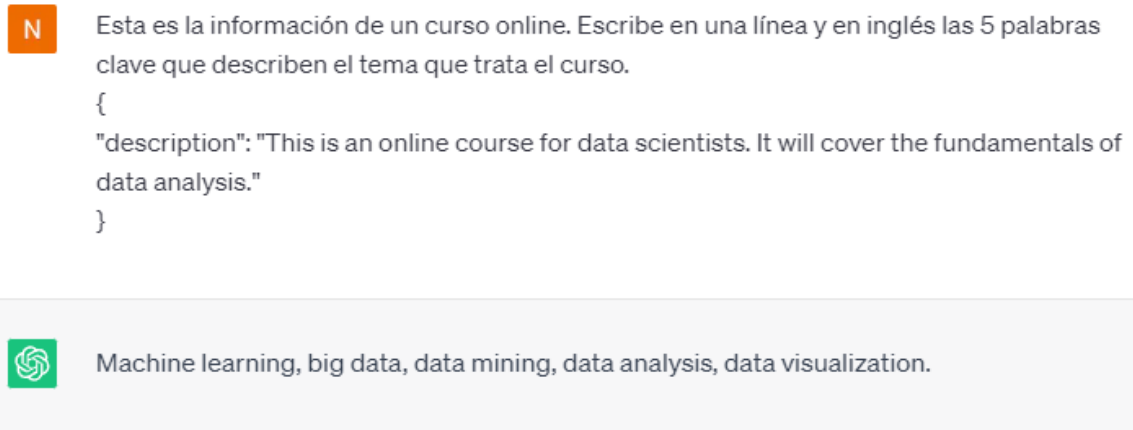
En la *Tabla 6* se pueden ver los resultados de la extracción de palabras clave de los textos objetivo.

Bases de datos con SQL Server	<b>SQL Server, Database Design, relational databases, SQL queries, Microsoft SQL Server</b>
Ciberseguridad. Protege tu información de Cibercriminales.	<b>Cybersecurity, Information Protection, Cybercrime, Data Privacy, Security Skills</b>
Angular PRO desde cero: El curso definitivo (Angular 8+)	<b>Angular, Web Development, Components, Routing, Forms.</b>
Aprende a programar jugando con Scratch	<b>Programming language, Scratch, Game Development, Computational thinking, Online course.</b>

*Tabla 6. Palabras clave de textos objetivo con ChatGPT*

Como bien se puede observar, *ChatGPT* no tiene dificultades a la hora de extraer las palabras clave de cada texto objetivo, pues todas las palabras extraídas describen el tema de cada curso. Además, como *ChatGPT* sabe que los textos representan cursos online, no devuelve ninguna palabra que tenga que ver con el concepto de aprendizaje, ya que se le ha indicado que las palabras deben describir el tema del curso, pero no el texto en sí mismo. De hecho, varias de las palabras que extrae no aparecen en el texto objetivo, sino que las genera *ChatGPT* al detectar el tema del curso.

Por otro lado, el comportamiento de esta técnica con textos más cortos también es positivo, pues la inteligencia artificial es capaz de generar las palabras con muy poca información, como se puede ver en la *Figura 15*.



*Figura 15. Palabras clave de texto corto con ChatGPT*

En definitiva, la extracción de palabras clave que describan el tema del texto mediante ChatGPT parece una manera muy eficaz y fiable de caracterizar los textos, por las siguientes razones:

- Las palabras extraídas representan el tema del curso en un 99% de los casos.
- ChatGPT entiende el contenido del texto, por lo que solamente devuelve palabras que describan su tema, y no el texto en sí.
- ChatGPT genera palabras clave de manera correcta aún cuando el texto es corto.

Sin embargo, la tecnología de OpenAI presenta dos inconvenientes:



- La API de ChatGPT es unas diez veces más cara que las de Google e IBM.
- Debido a la alta demanda, los servicios de ChatGPT pueden no funcionar en ciertos momentos.

### ***5.3.2.5 Conclusiones de las pruebas de caracterización.***

En conclusión, se tienen dos alternativas a la hora de caracterizar los textos.

Por un lado, *IBM* aporta una opción más económica y estable con su funcionalidad de clasificación. Sin embargo, presenta el inconveniente principal de que las caracterizaciones son imprecisas en un 20% de los casos. Esto puede suponer problemas de cara a los resultados, pero merece la pena investigar su potencial.

Por otro lado, *OpenAI* ofrece una alternativa muy potente y flexible con *ChatGPT*. A costa de un precio más alto, esta opción caracteriza los textos de manera muy precisa y fiable, por lo que sería la opción más segura.

Debido a la diferencia de coste entre las dos opciones, es interesante intentar cumplir con la funcionalidad con el modelo de *IBM* primero. Si no se consigue, la idea es pasar a la alternativa de *OpenAI*.

### **5.3.3 PRUEBAS DE COMPARACIÓN**

Habiendo terminado con las pruebas de caracterización, se procede a realizar las pruebas del segundo bloque: el bloque de comparación. En esta sección, el objetivo es comprobar de qué manera se pueden comparar las caracterizaciones anteriormente obtenidas, con el fin de que dos textos de mismo tema obtengan un alto grado de similitud. Para cumplir este objetivo, existen dos alternativas: *Spacy* y, de nuevo, *OpenAI*.

### 5.3.3.1 Pruebas de comparación con Spacy

Como se menciona en la *Sección 5.2.2*, Spacy tiene la capacidad de comparar palabras y frases mediante la generación de *embeddings*. Así pues, se debe comprobar su precisión a la hora de comparar las caracterizaciones obtenidas del primer bloque. Entonces, la idea es analizar la similitud que Spacy devuelve sobre diferentes caracterizaciones. Dos caracterizaciones muy parecidas deberían devolver una similitud alta. Dos caracterizaciones muy distintas deberían devolver una similitud muy baja. La similitud debe ser un porcentaje.

De esta manera, se han realizado pruebas con las caracterizaciones obtenidas con el modelo de clasificación de IBM. Por esto, el formato de las caracterizaciones es siempre el mismo: una categoría como las ya vistas en la *Tabla 3*. Por ejemplo: “*technology & computing/computing/computer networking*”.

En la *Tabla 7*, se muestra una de las pruebas realizadas. Esta consiste en calcular la similitud de varias categorías con la categoría “*technology & computing/computing/computer networking*”. Mediante una función en Python, se obtienen los embedding de Spacy de cada categoría, y se genera la similitud entre ellos empleando la propia librería de Spacy.

<b>Similitudes con “<i>technology &amp; computing/computing/computer networking</i>”</b>	
/technology & computing/computing/computer networking	<b>100%</b>
/technology & computing/computing/computer software and applications	<b>97.8%</b>
/technology & computing/computing/computer software and applications/databases	<b>96.8%</b>
/technology & computing/computing/computer software and applications/operating systems	<b>96.7%</b>
/technology & computing/computing/information and network security	<b>95.8%</b>

/technology & computing/computing/computer software and applications/digital audio	95.5%
/technology & computing/computing/internet/email	94%
/technology & computing/artificial intelligence	93.4%
/technology & computing/computing/data storage and warehousing	91.5%
/technology & computing/computing/internet/web hosting	91.2%
/technology & computing/computing/internet/internet of things	88.6%
/business and finance/business/business i.t.	68%

Tabla 7. Similitud entre categorías de IBM con Spacy

Como se puede observar, la similitud entre dos categorías exactamente iguales es, lógicamente, del 100%. Además, el orden de similitud de las categorías parece bastante acertado, pues las primeras tratan más sobre sistemas de redes y de bases de datos, mientras que las segundas tienden hacia el tema de internet, y la última categoría es la más enfocada a temas de empresa.

Sin embargo, la diferencia de similitud entre las categorías es muy baja, puesto que todas están por encima del 88% (menos la de empresa). Esto se debe a que todas esas categorías tienen muchas palabras en común con la que se están comparando: “/technology & computing/computing/computer”. Entonces, una forma de aliviar este problema es eliminando las palabras en común dentro de cada comparación. De esta manera, la similitud solamente se calcula sobre las palabras que son diferentes. En la *Tabla 8* se muestran los resultados aplicando esta nueva técnica.

<b>Similitudes con “technology &amp; computing/computing/computer networking”</b>	
/technology & computing/computing/computer networking	<b>100%</b>
/technology & computing/computing/computer software and applications	<b>96.8%</b>
/technology & computing/computing/computer software and applications/operating systems	<b>95.9%</b>
/technology & computing/computing/computer software and applications/databases	<b>95.5%</b>
/technology & computing/computing/computer software and applications/digital audio	<b>93.6%</b>
/technology & computing/computing/information and network security	<b>89.4%</b>
/technology & computing/artificial intelligence	<b>85.6%</b>
/technology & computing/computing/internet/email	<b>80.4%</b>
/technology & computing/computing/internet/web hosting	<b>80.1%</b>
/technology & computing/computing/data storage and warehousing	<b>78.8%</b>
/technology & computing/computing/internet/internet of things	<b>75.7%</b>
/business and finance/business/business i.t.	<b>68%</b>

*Tabla 8. Similitud entre categorías de IBM con Spacy eliminando palabras en común.*

Habiendo eliminado las palabras en común en cada comparativa, las confianzas se distribuyen en un intervalo más amplio, por lo que los resultados son más usables. Por tanto, se mejora la comparación de caracterizaciones.

En resumen, *Spacy* ofrece una comparación efectiva y sin coste. El único inconveniente que presenta es que las similitudes pueden ser demasiado altas, y que el tiempo de respuesta es elevado cuando se trabaja con muchos elementos. A pesar de esto, sigue siendo una alternativa válida para el bloque de comparación.

### 5.3.3.2 Pruebas de comparación con OpenAI

Al igual que *Spacy*, *OpenAI* permite generar *embeddings* de palabras o frases, como se demuestra en la *Sección 2.2*. La diferencia reside principalmente en el modelo que se usa para generarlos y en el formato de los vectores resultantes. También existe una diferencia en el cálculo de la similitud, pues ahora se hace mediante la distancia de coseno (ver *Sección 2.2*). Por tanto, la similitud entre dos caracterizaciones se observa a partir de la distancia que hay entre ellas. A menor distancia, más parecidas.

Para comprobar el funcionamiento de estas comparaciones, esta vez se realizan las pruebas sobre las caracterizaciones generadas por *ChatGPT* en vez de las de *IBM*. Esto quiere decir que se introducen caracterizaciones en formato de cinco palabras clave en lugar de etiquetas de categorías.

En la *Tabla 9*, se muestran las distancias entre la caracterización en forma de palabras clave “*Cybersecurity, Information Security, Data Protection, Threats, Protection Techniques*” y otras caracterizaciones.

<b>Distancias con “<i>Cybersecurity, Information Security, Data Protection, Threats, Protection Techniques.</i>”</b>	
Cybersecurity, Information Security, Data Protection, Threats, Protection Techniques	0

Information Security Management, ISO 27001, Security Standards, Cybersecurity, IT Certifications.	<b>0.11</b>
OSINT, Advanced Digital Espionage, Cyberpatrol, Internet Investigations, Privacy Protection.	<b>0.14</b>
DevOps, General Concepts, Three Ways, Implementation, Metrics.	<b>0.18</b>
Machine Learning, Production Engineering, Model Deployment, MLOps, Model Monitoring	<b>0.2</b>
Programming language, Scratch, Game Development, Computational thinking, Online course.	<b>0.21</b>
Angular, Web Development, Components, Routing, Forms.	<b>0.22</b>
Model deployment, scalable infrastructure, workflow automation, MLOps practices, model monitoring.	<b>0.24</b>
TikTok, Marketing, Promote, Business, Course.	<b>0.24</b>
presentations, storytelling, communication, visual, public speaking.	<b>0.25</b>
Recruitment, selection, interview, human resources, hiring.	<b>0.25</b>

*Tabla 9. Distancias entre palabras clave con OpenAI*

Como se puede ver, la distancia entre dos caracterizaciones exactamente iguales es, lógicamente, nula. Además, las caracterizaciones que están a menor distancia son aquellas formadas por palabras relacionadas con la ciberseguridad, lo que cuadra perfectamente con el caso.

Tal y como se enseña en la *Sección 2.2*, se puede representar estas 11 caracterizaciones en un espacio de dos dimensiones usando el algoritmo de *TSNE*.

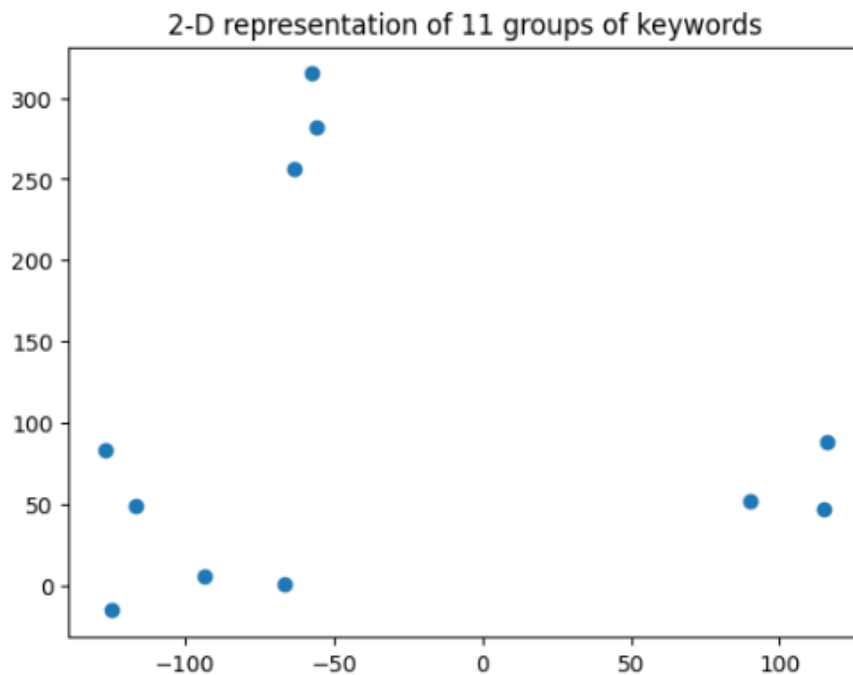


Figura 16. Representación de palabras clave en dos dimensiones con *OpenAI*

En la *Figura 16*, se observan tres grupos muy diferenciados de caracterizaciones. Los dos grupos de tres elementos corresponden a las tres primeras y tres últimas caracterizaciones de la *Tabla 9*. El primer grupo trata sobre ciberseguridad, mientras que el segundo contiene palabras relacionadas con el mundo social. Por otro lado, se observa un grupo de cinco elementos, que representan las cinco caracterizaciones del centro de la *Tabla 9*. Estas cinco caracterizaciones tratan el tema de la programación y del desarrollo de Software.

En definitiva, la generación de embeddings con *OpenAI* sobre las caracterizaciones devueltas por *ChatGPT* en formato de palabras clave da un buen resultado. Además, el coste de esta tecnología es reducido, por lo que no supone una desventaja frente a la funcionalidad gratuita de *Spacy*.

### 5.3.3.3 Conclusiones de las pruebas de comparación

En conclusión, se tienen dos alternativas a la hora de comparar caracterizaciones de los textos.

Por un lado, *Spacy* ofrece una generación de *embeddings* gratuita que es capaz de comparar satisfactoriamente las categorías generadas por *IBM*. El inconveniente que presenta es que el cálculo de la similitud es lento, lo que puede ser problemático.

Por otro lado, *OpenAI* también permite la generación de embeddings bajo un reducido coste. Usando el mismo modelo que la potente tecnología de *ChatGPT*, puede transformar grupos de palabras clave en vectores, y calcular la distancia entre ellos de manera muy rápida. Además, estos vectores se pueden representar en un espacio bidimensional, lo que ayuda a visualizar los resultados.

En resumen, tanto *Spacy* como *OpenAI* presentan alternativas válidas ante la solución del bloque de comparación de caracterizaciones, por lo que no se debe descartar ninguna.

### 5.3.4 CONCLUSIONES DE LA FASE DE SELECCIÓN DE TECNOLOGÍAS

Tras haber realizado las pruebas de caracterización y de comparación, surgen dos opciones de selección de tecnologías para el desarrollo del modelo final, como se puede ver en la *Tabla 10*.

	Bloque de caracterización	Bloque de comparación
Opción 1	Clasificación con <i>IBM</i>	Embeddings con <i>Spacy</i>
Opción 2	Palabras clave con <i>ChatGPT</i>	Embeddings con <i>OpenAI</i>

*Tabla 10. Opciones de diseño del modelo final*



## 5.4 FASE DE DESARROLLO DEL MODELO

A raíz de la fase de selección de tecnologías, se llega a esta fase de desarrollo con dos opciones de diseño, como se muestra en la *Tabla 10*. Así pues, deben desarrollarse ambos modelos para ver cuál resulta ser más efectivo. Este es el objetivo de esta fase, con el fin de acabar con un modelo que dé la funcionalidad completa al proyecto.

Siguiendo con el mismo contexto que el descrito en la *Sección 5.3.1*, la idea es desarrollar un modelo que sea capaz de recomendar una lista de cursos (textos objetivo) a partir del contenido de una prueba sobre un tema concreto (texto de entrada). En esta fase, se deben conectar los bloques de caracterización y de comparación, para acabar montando el sistema al completo.

Con el fin de obtener unos resultados más realistas y completos, los modelos desarrollados se pondrán a prueba con una gran variedad de textos de entrada. Anteriormente, todos los textos de entrada han sido de ámbito tecnológico, pero en esta fase se va a trabajar con otros sectores como el comercial, el industrial, social, etc.

### 5.4.1 MODELO 1: IBM CON SPACY

En este primer modelo, se emplea la funcionalidad de clasificación de *IBM Watson* para el bloque de caracterización, y la generación de *embeddings* de *Spacy* para el bloque de comparación. El bloque de selección consiste simplemente en una función que ordena las similitudes que le llegan y devuelve los  $n$  textos objetivos con similitud más alta.

Entonces, empleando la estructura descrita en la *Sección 5.1.2*, el diagrama completo de este modelo queda como se muestra en la *Figura 17*.

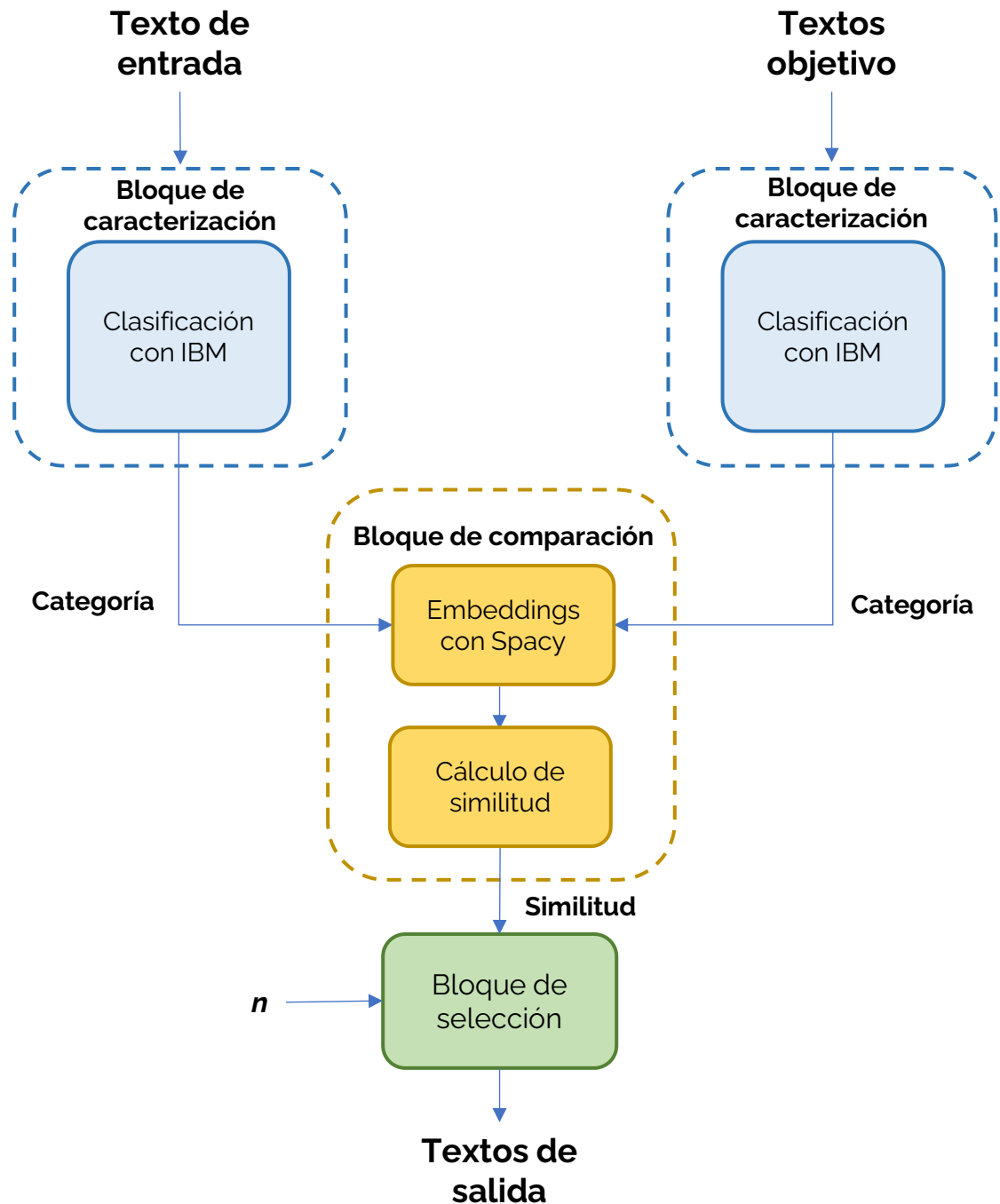


Figura 17. Diagrama del Modelo 1

Es importante notar que, aunque no se vea reflejado en el diagrama, la caracterización de los textos objetivo no debe hacerse en cada ejecución del modelo. Estas caracterizaciones pueden almacenarse desde el principio para ahorrar costes y tiempo, evitando llamadas

innecesarias a la *API* de *IBM*. También se podrían almacenar los *embeddings* de los textos objetivo generados por *Spacy* pero, como su generación no tiene coste y es muy rápida, no es algo indispensable.

El proceso completo que sigue el Modelo 1 es siguiente:

- Se manda el texto de entrada a la *API* de *IBM* y se obtiene su caracterización en forma de categoría.
- Se genera el *embedding* de la caracterización del texto de entrada usando *Spacy*.
- Se generan los *embeddings* de las caracterizaciones almacenadas de todos los textos objetivo.
- Se calcula la similitud entre el *embedding* del texto de entrada y los *embeddings* de los textos objetivo.
- Se introducen las similitudes en el bloque de selección, que devuelve los  $n$  textos objetivo con la mayor similitud.

#### 5.4.2 MODELO 2: CHATGPT CON OPENAI

En este segundo modelo, se utiliza *ChatGPT* para generar palabras clave que describan el tema de los textos dentro del bloque de caracterización, y los *embeddings* de *OpenAI* junto con la distancia de coseno para el bloque de comparación. Por su parte, el bloque de selección consiste en una función que recibe las distancias y devuelve aquellas que no superan un umbral de distancia máxima.

Empleando de nuevo la estructura de la *Sección 5.1.2*, el diagrama de bloques del Modelo 2 se puede ver en la *Figura 18*.

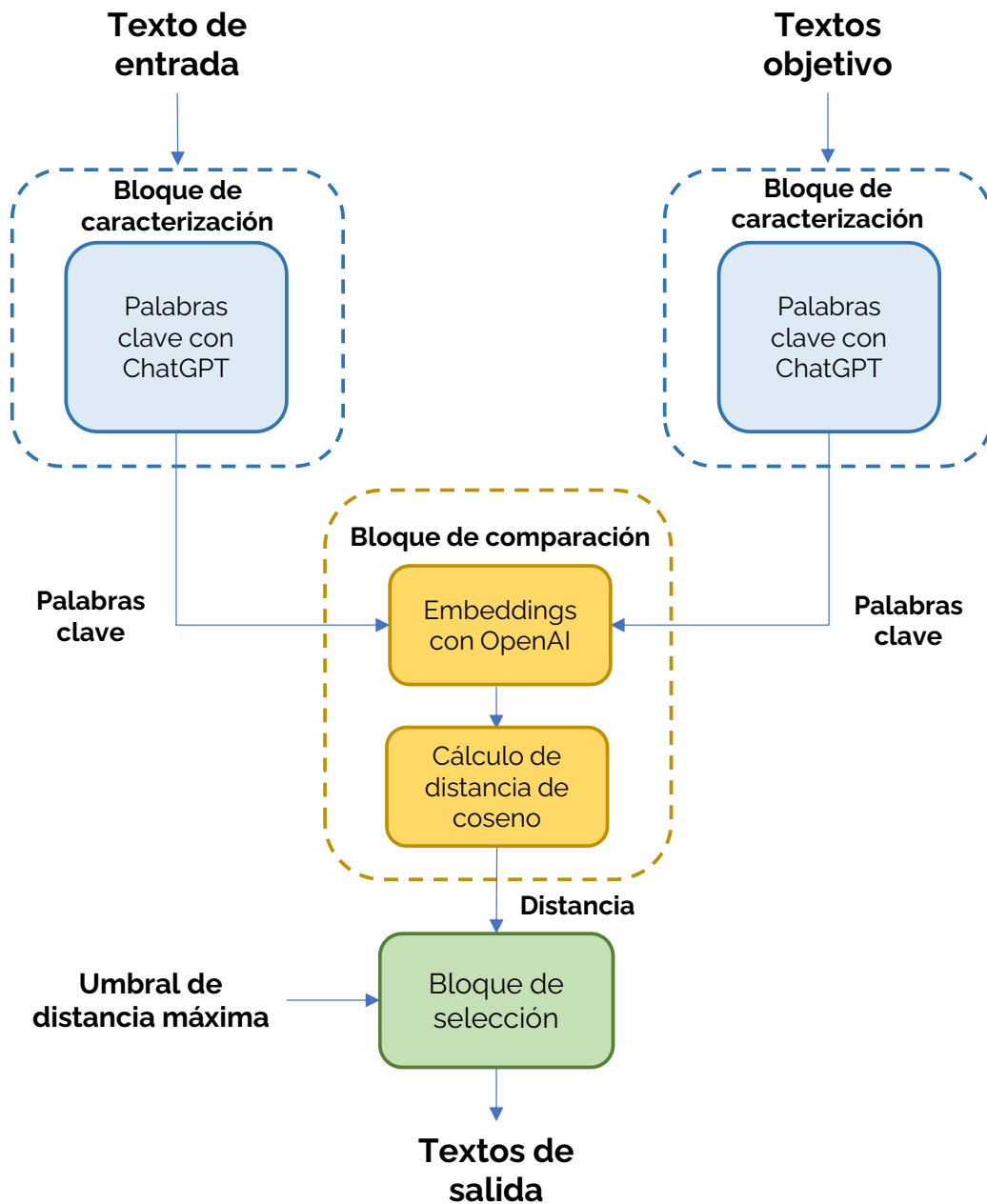


Figura 18. Diagrama del Modelo 2

Al igual que en el Modelo 1, las caracterizaciones de los textos objetivo pueden almacenarse para que solamente haga falta obtener las palabras clave una vez, reduciendo enormemente el coste y el tiempo de ejecución del proceso.

Además, se ha incluido una pequeña mejora en este modelo, habiendo cambiado el parámetro que utiliza el bloque de selección para devolver los textos de salida. Ahora, este parámetro es un umbral de distancia máxima, que ningún texto de salida puede superar. Esto hace que no se generen asociaciones falsas por falta de textos objetivo, como sí que podía ocurrir en el Modelo 1.

El proceso completo que sigue el Modelo 2 es el siguiente:

- Se manda el texto de entrada a la *API* de *ChatGPT* y se obtiene su caracterización en forma de 5 palabras clave que definen el tema del texto.
- Se genera el *embedding* de esta caracterización usando *OpenAI*.
- Se generan los *embeddings* de las caracterizaciones almacenadas de todos los textos objetivo usando *OpenAI*.
- Se calcula la distancia de coseno entre el *embedding* del texto de entrada y los *embeddings* de todos los textos objetivo.
- Se introducen las distancias en el bloque de selección, que devuelve solamente aquellos textos objetivo que estén por debajo del umbral de distancia máxima.

Una desventaja que presenta este modelo es que la *API* de *ChatGPT* que se usa para generar las palabras clave puede no llegar a responder por sobrecarga debido a su alta demanda. Esto no causa problemas para la caracterización del texto de entrada, pues se trata de un único texto. El inconveniente aparece a la hora de generar las palabras de los miles de textos objetivo por la cantidad de peticiones que hay que hacer a la *API*. Por esto, se ha diseñado un programa que recorre la lista de textos objetivo y los va mandando uno a uno. En cada iteración del bucle, las nuevas palabras clave se añaden a un archivo comprimido que va acumulando todas las caracterizaciones. Si en alguna iteración la *API* deja de responder, el programa se detiene y, al reanudarlo, se retoma el proceso desde el último texto objetivo que

se caracterizó. Esto permite generar las palabras clave de los textos de manera cómoda sin que los fallos de la *API* corrompan el sistema.

## **Capítulo 6. ANÁLISIS DE RESULTADOS**

Introduciendo una gran variedad de textos de entrada en ambos modelos, se puede comprobar su comportamiento tratando contenido de distintos ámbitos. Por tanto, en las siguientes figuras se exponen las recomendaciones de cursos generadas por cada modelo, a partir de textos de entrada de diferentes sectores profesionales. Estos resultados se han obtenido a partir de un banco de 900 textos objetivo en español, todos representando cursos online reales de diversos temas.

### ***6.1 RESULTADOS DEL MODELO 1***

En cada una de las tablas, el encabezado indica el título del texto de entrada, es decir, el título de la prueba. En la primera columna, se muestra el título del curso recomendado y, en la segunda, la similitud calculada con el texto de entrada. No se muestran todos los resultados obtenidos, sino los suficientes para sacar conclusiones.

## Redes

	course	similarity
0	Cisco CCNA Fundamentos de Networking para Redes IP	0.7529
1	Principios de radioenlaces para WISP	0.6507
2	Amazon AWS: Curso Completo Arquitecto Soluciones Certificado	0.5995
3	Aprende Redes desde Cero: Curso Completo	0.5970
4	Configuración básica de switches Cisco para principiantes	0.5968

## Routing

	course	similarity
0	Configuración básica de switches Cisco para principiantes	0.7170
1	Cisco CCNA Fundamentos de Networking para Redes IP	0.6625
2	Cisco CCNA 200-301 - Practicas de configuracion en español	0.6110
3	AD DS, DNS y DHCP Completo de cero a experto!!	0.6003
4	Principios de radioenlaces para WISP	0.5903

## Arquitectura Software

	course	similarity
0	Administración PostgreSQL: Técnicas de Backup y Recuperación	0.7217
1	Análisis en Microsoft Excel: Power Pivot, DAX y Power Query.	0.6727
2	Administración de Base de Datos Con SQL Server	0.6682
3	AWS Certified SysOps Administrator Associate SOA-C02 Español	0.6433
4	Base de datos con SQL Server administra y crea 2022	0.6237

*Figura 19. Resultados del Modelo 1 en el ámbito tecnológico*



## Dirección Comercial

	course	similarity
0	Programa de Innovación Empresarial	0.8554
1	Fundamentos de contabilidad financiera	0.8150
2	Comercio Exterior y Logística Internacional	0.7764
3	Fusiones y Adquisiciones: Acoples Estratégicos y Valoración	0.7551
4	Análisis de Marketing: Estrategias y Análisis de Precios	0.7451

## Técnicas de Ventas

	course	similarity
0	Consultor de Marketing y Publicidad	0.7179
1	Community Manager & Social Media para PRINCIPIANTES	0.6330
2	Comunicación Avanzada: Escucha Activa, Empatía y Asertividad	0.6242
3	Crea un Embudo de Ventas desde Cero para tu Negocio Digital	0.5927
4	Japonés para Principiantes: Idioma y Cultura	0.5626

## Técnicas de Atención al Cliente

	course	similarity
0	Curso Consultor Seguridad SAP - Seguridad Informatica	0.7215
1	Hacking Wifi Profesional. Realiza Auditorias sobre Seguridad	0.6746
2	Gestión de Seguridad de la Información - ISO 27001	0.6217
3	Curso de Introducción a la Seguridad Informática desde Cero!	0.6168
4	Certificación Cisco CCNA 200-301: Introducción a las Redes.	0.6002

*Figura 20. Resultados del Modelo 1 en el ámbito comercial*

## Análisis Cuantitativo

	course	similarity
0	Bolsa de Valores: Análisis Técnico con Fibonacci	0.7921
1	Taller de Contabilidad Computarizada	0.7558
2	Análisis financiero corporativo	0.7437
3	Inteligencia Financiera: Guía completa Finanzas Personales	0.7330
4	Invertir en Bolsa: Introducción al Análisis Técnico	0.7312

## Análisis Financiero

	course	similarity
0	Análisis financiero corporativo	0.9332
1	Gestión de Riesgos Financieros	0.7842
2	Evaluación Financiera de Proyectos de Inversión	0.7599
3	Invertir en Bolsa: Introducción al Análisis Técnico	0.7055
4	Taller de Contabilidad Computarizada	0.6965

## Mercados Financieros

	course	similarity
0	Gestión de Riesgos Financieros	0.8113
1	Evaluación Financiera de Proyectos de Inversión	0.7486
2	Análisis financiero corporativo	0.7480
3	Inteligencia Financiera: Guía completa Finanzas Personales	0.7084
4	Bolsa de Valores: Análisis Técnico con Fibonacci	0.6884

Figura 21. Resultados del Modelo 1 en el ámbito financiero

## Relaciones Laborales

	course	similarity
0	Gestión de Nóminas y Seguros Sociales	0.6578
1	Implementación y Cumplimiento del Estándar PCI DSS	0.6052
2	Análisis financiero corporativo	0.6010
3	Modelos Financieros en Excel para la Valoración de Empresas	0.5917
4	Evaluación Financiera de Proyectos de Inversión	0.5858

## Prevención de Riesgos Laborales

	course	similarity
0	CALMA: Curso Avanzado para el Manejo Exitoso de la Ansiedad	0.7363
1	Curso de Preparación de Agile Tester Extension Certification	0.7254
2	People Analytics con Power BI: Análisis en Recursos Humanos	0.6906
3	Angular: Pruebas unitarias con Jasmine y Karma	0.6624
4	Angular: Pruebas unitarias con Jest	0.6435

## Técnicas de RRHH

	course	similarity
0	LinkedIn Sales Navigator: La herramienta para ventas B2B	0.5710
1	Excel Avanzado: Power Pivot y Power BI + CERTIFICADO	0.5031
2	Aumenta tu Productividad y Alcanza tus Metas	0.3989
3	Excel Completo - Desde Principiante Hasta Avanzado	0.3988
4	Comercio Internacional & Logística para Importar y Exportar	0.3896

Figura 22. Resultados del Modelo 1 en el ámbito de RRHH

## Edificación

	course	similarity
0	Análisis de Call Center con Business Intelligence y Power BI	0.6244
1	BPMN para Analistas y Arquitectos de Negocios	0.5999
2	El Analista de Inteligencia Empresarial	0.5825
3	Business Intelligence y Análisis de Datos con QLIK SENSE	0.5676
4	Transformación Digital para Empresas y Profesionales.	0.5475

## Transporte

	course	similarity
0	Implementador Sistema de Gestión de la Calidad ISO 9001:2015	0.5647
1	Curso de Gestión de Equipos y Trabajo en Remoto	0.5522
2	SAP MM Datos Maestros; Materiales, Servicios y Proveedores.	0.5131
3	Curso de Industria 4.0	0.4937
4	Análisis de Estados Financieros: Crucial Business Analytics	0.4874

## Urbanismo

	course	similarity
0	Máster en Gestión y Mejora de Procesos	0.4484
1	Administración y logística en la cadena de suministro	0.4187
2	MÁSTER en Arquitectura Ecológica, Sostenible y Bioclimática.	0.4069
3	Cadena logística y transporte internacional	0.3953
4	Comercio Exterior y Logística Internacional	0.3862

*Figura 23. Resultados del Modelo 1 en el ámbito de Ingeniería*

Analizando las tablas anteriores, se sacan las siguientes conclusiones sobre los resultados del Modelo 1 en cada uno de los ámbitos. La validez de los resultados viene dada por la relación entre el texto objetivo recomendado y el texto de entrada y, en un segundo plano, la similitud calculada dentro de cada asociación.

En el ámbito tecnológico (*Figura 19*), el modelo se comporta satisfactoriamente. Los cursos recomendados son adecuados a los textos de entrada, aunque algunas similitudes son bajas (inferiores a 60%).

En el ámbito comercial (*Figura 20*), los resultados muestran algunas inconsistencias. Por ejemplo, la recomendación del curso de japonés para el texto de entrada sobre técnica de ventas no es adecuada. Además, las cinco recomendaciones sobre la última evaluación (Técnicas de atención al cliente) son erróneas. Los resultados del primer texto de entrada (Dirección comercial) son correctos.

En el ámbito financiero (*Figura 21*), los resultados son positivos. El modelo es capaz de asociar bien cada texto de entrada con sus textos objetivo de tema más cercano. Por tanto, las recomendaciones son válidas y las similitudes son altas.

En el ámbito de Recursos Humanos (*Figura 22*), el modelo muestra debilidad, pues solamente 2 de las 15 asociaciones puede considerarse válidas. El resto de cursos recomendados no tratan un tema similar al del texto de entrada.

En el ámbito de ingeniería (*Figura 23*), los resultados también son negativos. Las asociaciones generadas por el sistema no son correctas, pues se relacionan exámenes sobre ingeniería con cursos de diferentes temas.

En conclusión, el Modelo 1 se comporta bien cuando trata textos de temas tecnológicos o financieros. Cuando se sale de estos dos ámbitos, los resultados comienzan a fallar, haciendo que no sea un modelo viable en esos casos. Puesto que Spacy ha demostrado funcionar correctamente en el bloque de comparación (ver *Sección 5.3.3.1*), el problema del Modelo 1 reside en el bloque de caracterización en el que se emplea la funcionalidad de clasificación de *IBM*.

## **6.2 RESULTADOS DEL MODELO 2**

De la misma manera que con los resultados del Modelo 1, a continuación, se presentan una serie de tablas que muestran las recomendaciones generadas para textos de entrada de diferentes ámbitos.

Esta vez, la segunda columna indica la distancia medida entre el embedding de cada texto objetivo con el texto de entrada que aparece encima de cada tabla. Además, se ha ajustado el umbral de distancia máxima del bloque de selección a un valor de 0.16. Por esto, hay algunos textos de entrada que generan hasta 10 asociaciones, mientras que otros no generan ninguna.

## Redes

	course	distance
0	Seguridad informática para Teléfonos. Protege tus Datos.	0.131
1	Cisco CCNA 200-301 en Español + Simulador de Preguntas !	0.134
2	Redes de datos y detección de fallas	0.136
3	Certificación CCNA Versión 200-301. Aprende CCNA 2 y 3.	0.138
4	Certificación Cisco CCNA 200-301: Introducción a las Redes.	0.138
5	Fundamentos de Redes. Como se realizan las Comunicaciones.-	0.139
6	Cisco CCNA 200-301 en español	0.14
7	Como configurar una red de datos desde Cero de forma fácil	0.142
8	Curso de redes en Microsoft Azure (IaaS y PaaS)	0.142
9	Fundamentos del técnico en redes de fibra óptica	0.143

## Routing

	course	distance
0	Cisco Curso de direccionamiento IP y Subnetting - CCNA	0.115
1	Cisco CCNA Fundamentos de Networking para Redes IP	0.122
2	Fundamentos Cisco Networking Parte 1	0.125
3	Cisco CCNA 200-301 en español	0.128
4	Cisco CCNA 200-301 - Practicas de configuracion en español	0.132
5	Como configurar una red de datos desde Cero de forma fácil	0.134
6	Certificación CCNA Versión 200-301. Aprende CCNA 2 y 3.	0.134
7	Cisco CCNA 200-301 en Español + Simulador de Preguntas !	0.137
8	Curso Cisco BGP nivel CCNP Encor by SeaCCNA	0.139
9	Fundamentos de Redes. Como se realizan las Comunicaciones.-	0.141

Figura 24. Resultados del Modelo 2 en el ámbito tecnológico.

## Marketing de Producto

	course	distance
0	Consultor de Marketing y Publicidad	0.092
1	Análisis de Marketing: Estrategias y Análisis de Precios	0.116
2	Curso de Branding - Marca, Logotipo, Naming, Marketing, Logo	0.116
3	Estrategia de Marketing y Marketing Digital 2023 - desde 0	0.127
4	Branding y Neuromarketing digital, la ciencia de vender	0.136
5	Marketing Digital: Principios que son Claves y Nadie Enseña	0.14
6	Automatiza tu Marketing Digital. Gana mas Tiempo libre.	0.141
7	Máster de Marketing Digital - 12 cursos en 1	0.142
8	Neuromarketing: Neurociencia aplicada para crecer tu negocio	0.144
9	Maestría en ventas	0.145

## Desarrollo de Negocio

	course	distance
0	Estrategia de Marketing y Marketing Digital 2023 - desde 0	0.109
1	Consultor de Marketing y Publicidad	0.11
2	Curso Completo Paso a Paso de Marketing de Afiliados Exitoso	0.116
3	Automatiza tu Marketing Digital. Gana mas Tiempo libre.	0.121
4	Marketing Digital: Secretos para Acelerar tus Ventas!	0.121
5	Máster de Marketing Digital - 12 cursos en 1	0.123
6	Marketing Digital - Generar Leads y Convertirlos en Ventas	0.127
7	Máster en Marketing Digital - Posiciona tu negocio en la web	0.128
8	Embudo de Ventas: Cómo Conseguir Clientes Referidos	0.132
9	Curso de Branding - Marca, Logotipo, Naming, Marketing, Logo	0.132

*Figura 25. Resultados del Modelo 2 en el ámbito comercial.*



## Análisis Cuantitativo

	course	distance
0	Estadística para Data Science y análisis de negocios 2023	0.099
1	Domina las Matemáticas Financieras	0.104
2	Estadística para todos. Análisis de datos y Toma de decisión	0.105
3	Econometría con STATA desde Básico hasta Avanzado	0.113
4	Análisis financiero corporativo	0.116
5	Estadística descriptiva e inferencial con R	0.117
6	Aprende matemáticas desde cero - Cálculo Diferencial	0.117
7	Análisis de Estados Financieros	0.121
8	Finanzas para No Financieros	0.121
9	Taller de Contabilidad Computarizada	0.123

## Análisis Financiero

	course	distance
0	Análisis financiero corporativo	0.06
1	Valoración de Empresas: Descubre Cuánto Pagar por un Negocio	0.081
2	Análisis de Estados Financieros	0.088
3	Inversión en Acciones y Bolsa de Valores (De 0 a Avanzado)	0.094
4	Planificación Financiera: Estados Financieros Presupuestados	0.096
5	Modelos Financieros en Excel para la Valoración de Empresas	0.099
6	Análisis de Estados Financieros: Crucial Business Analytics	0.104
7	Taller de Contabilidad Computarizada	0.113
8	Fusiones y Adquisiciones: Acoples Estratégicos y Valoración	0.113
9	Introducción a las Finanzas	0.116

*Figura 26. Resultados del Modelo 2 en el ámbito financiero.*

## Relaciones Laborales

	course	distance
0	Gestión de Nóminas y Seguros Sociales	0.081
1	Protege a tu empresa de demandas laborales (México)	0.146
2	El ABC de Recursos Humanos	0.149
3	Comercio Exterior y Logística Internacional	0.155

## Prevención de Riesgos Laborales

	course	distance
0	BIOSEGURIDAD LABORAL COVID 19 - CONSEJOS Y GUIAS PRACTICAS	0.109
1	Formación en prevención de riesgos laborales - COVID-19	0.116
2	ISO 45001:2018, Sistemas de Gestión de Seguridad y Salud	0.123
3	Aplicación de la norma NOM035	0.127
4	Higiene y seguridad alimentaria: diseña tu sistema APPCC	0.127
5	NOM-035. Factores de riesgo psicosocial en el trabajo	0.135
6	Seguridad Psicológica para Equipos Altamente Efectivos	0.136
7	Prevención y Protección contra Incendios en el Trabajo	0.137
8	Protege a tu empresa de demandas laborales (México)	0.138
9	Protocolo COVID-19 para Trabajo Seguro.	0.14

## Técnicas de RRHH

	course	distance
0	Gestión del Talento en las Organizaciones	0.058
1	El ABC de Recursos Humanos	0.07
2	Reclutamiento, entrevista y selección de personal.	0.079
3	Gestión Estratégica del Capital Humano	0.08
4	Curso Completo de Reclutamiento y Selección de Personal	0.091
5	Selección IT y Marketing para Reclutamiento	0.095
6	People Care IT	0.096
7	Programa Avanzado Talento Humano 3.0   RRHH	0.105
8	No es un curso más de recruiting IT	0.111
9	Onboarding - Guía práctica para transformar tu empresa	0.127

Figura 27. Resultados del Modelo 2 en el ámbito de RRHH.

## Edificación

	course	distance
0	Materiales. Acabados y Texturas Arquitectura e Interiorismo	0.108
1	Aprende Revit: orientado a la arquitectura	0.153
2	MÁSTER Arquitectura de Interiores, Interiorismo y Decoración	0.158

## Transporte

	course	distance
0	Cadena logística y transporte internacional	0.128
1	MATLAB para Ingeniería Civil	0.133
2	Comercio Exterior y Logística Internacional	0.14
3	Curso Completo SAP MM Desde Cero	0.145
4	Experto en Diseño y Planificación Urbana y del Paisaje	0.145
5	Liderazgo: Gerencia en Tecnología / Ingeniería 2023	0.146
6	Optimización y Logística de Almacenes Inventarios y Stocks	0.149
7	Comercio Internacional & Logística para Importar y Exportar	0.154
8	Curso de inglés para profesionales - Business English Basics	0.158
9	Aprende Terraform con AWS desde cero (2021) v1.0 compatible	0.159

## Urbanismo

	course	distance
0	Experto en Diseño y Planificación Urbana y del Paisaje	0.125

Figura 28. Resultados del Modelo 2 en el ámbito de ingeniería.

Analizando las tablas anteriores, se obtienen las siguientes conclusiones sobre los resultados del Modelo 2. Al igual que en el Modelo 1, la validez de los resultados viene dada por la relación entre el tema de cada curso recomendado y el tema de su texto de entrada.

Empezando por el ámbito tecnológico (*Figura 24*), el modelo se comporta de manera muy satisfactoria. Debido a la gran abundancia de textos objetivo de tema tecnológico, muchos quedan a una distancia inferior al umbral de distancia máxima, por lo que se recomiendan muchos cursos, siendo todos ellos válidos.

En el ámbito comercial (*Figura 25*), ocurre lo mismo que en el tecnológico. Se recomiendan muchos cursos, ante la gran cantidad de textos objetivo que tratan el tema de marketing. Observando las asociaciones, todas relacionan cursos del tema del texto de entrada, por lo que son válidas.

En el ámbito financiero (*Figura 26*), sucede lo mismo que en los dos casos anteriores. Muchos cursos caen bajo el umbral de distancia máxima, pero todos lo hacen correctamente. Una vez más, el Modelo 2 genera asociaciones acertadas.

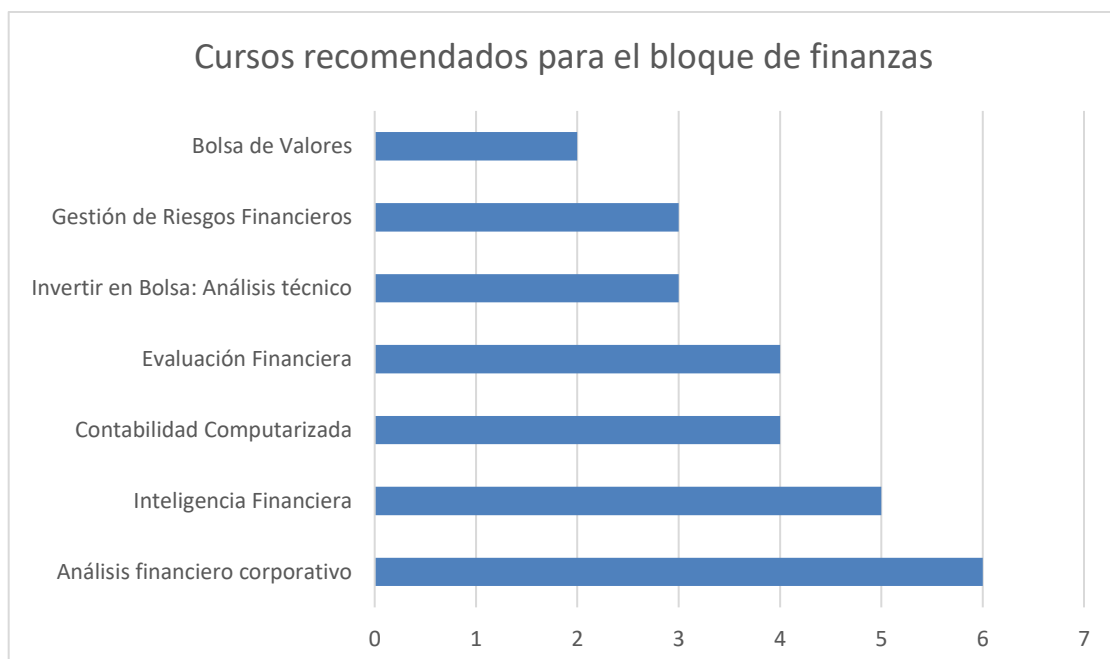
En el ámbito de Recursos Humanos (*Figura 27*), se observa un caso ligeramente diferente pues, en el primer texto de entrada (Relaciones laborales), el Modelo 2 devuelve solo 5 textos objetivo. Este es un ejemplo de caso en el no hay muchos textos objetivo que casen con el de entrada, es decir, el sistema solo encuentra 5 cursos que tratan el tema de Relaciones Laborales. Además, el último de ellos no se relaciona correctamente y está a una distancia de 0.155, cerca del umbral máximo. Este error se podría evitar ajustando el umbral de distancia máxima a 0.154.

Finalmente, en el ámbito de ingeniería (*Figura 28*), ocurre otro caso interesante. Para los textos de entrada sobre edificación y urbanismo, el modelo devuelve pocos textos objetivo, pues existen pocos cursos dentro del sistema que traten esos temas. A pesar de ello, los que devuelve son válidos. Concretamente, para el texto de entrada sobre urbanismo, el modelo solo recomendando un curso, que efectivamente es el único de todos los textos objetivo que trata sobre urbanismo, lo que demuestra precisión. Para el texto de entrada sobre transporte,

existen más textos objetivo relacionados con ese tema, por lo que el modelo recomienda diez cursos. Las dos últimas recomendaciones no son del todo válidas, pero esto se podría ajustar fácilmente con el umbral máximo de distancia.

En definitiva, los resultados del Modelo 2 son muy positivos en todos los ámbitos. Las pocas asociaciones erróneas se pueden eliminar reduciendo el umbral de distancia máxima, por lo que la precisión del modelo es casi del 100%. Tanto el bloque de caracterización como el bloque de comparación cumplen sus funciones, por lo que el diseño no presenta ninguna debilidad importante.

Para analizar de manera más completa los resultados del Modelo 2 sin tener que ir observando los textos devueltos por el sistema para cada uno de los textos de entrada, se ha realizado también un análisis por ámbitos. De esta manera, se escoge un ámbito que englobe varios textos de entrada y se cuentan cuántas veces se recomienda cada texto objetivo. Esto permite obtener una visión rápida y general de cómo se comporta el sistema en ese ámbito concreto. En la *Figura 29* se muestran los cursos recomendados para el bloque de finanzas, así como el número de veces que se ha recomendado cada curso.



*Figura 29. Cursos recomendados para el bloque de finanzas*

### **6.3 CONCLUSIONES SOBRE LOS RESULTADOS**

Tras analizar los resultados, se concluye rotundamente que el Modelo 2 es muy superior al Modelo 1, pues se comporta mejor en todos los ámbitos. El Modelo 1 se ve muy limitado por su diseño del bloque de caracterización empleando la clasificación de *IBM*. Los resultados en el ámbito tecnológico son buenos porque las categorías de *IBM* están más enfocadas en ese sector, pero, precisamente por esa razón, el Modelo 1 no cumple con los requisitos en el resto de ámbitos. Por tanto, a pesar del coste adicional del Modelo 2, merece la pena usarlo debido a sus grandes capacidades.

Por otra parte, para reafirmar las capacidades del Modelo 2, se ha puesto a prueba con un banco de 10000 textos objetivo con cursos de todo tipo de ámbitos y en diferentes idiomas. En este contexto, los resultados obtenidos son igual de satisfactorios, pero a mucha mayor escala.

Así pues, se concluye que el sistema diseñado con el Modelo 2 cumple con todos los requisitos descritos en la *Sección 5.1*, por lo que se ha encontrado una solución a la cuestión del proyecto.

## Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

### 7.1 CONCLUSIONES

En conclusión, mediante la realización de este proyecto, se han cumplido los dos siguientes objetivos.

Por un lado, se ha llevado a cabo un análisis de muchas de las tecnologías más relevantes dentro del mundo del procesamiento del lenguaje natural. Se han estudiado sus fortalezas y sus debilidades, y la utilidad de cada una dentro del concepto de asociación de textos. Se han realizado pruebas con las *API* de *Google Cloud*, *IBM* y *OpenAI*, así como con librerías de *Python* como *Spacy*. También se ha comprendido el concepto de los *embeddings*, y su utilidad y capacidad para comparar textos aplicando medidas vectoriales como la distancia de coseno. Por tanto, con este proyecto se ha cumplido una labor de investigación sobre la cuestión del *NLP*.

Por otro lado, se ha desarrollado un modelo que satisface todos los requerimientos del proyecto. Empleando las últimas tecnologías de *OpenAI*, este modelo es capaz de asociar textos muy distintos que compartan un mismo tema, sin importar su formato o estilo, y sin requerir ninguna fase de entrenamiento. Además, el sistema desarrollado propone un acercamiento distinto a los programas de recomendación, en el que la inteligencia artificial interpreta el contenido de los elementos en lugar de utilizar un algoritmo ajeno al lenguaje para generar las asociaciones. Todo esto supone una nueva aportación al mundo del *NLP* que, una vez más, demuestra su potencial para automatizar tareas humanas.

En definitiva, este trabajo cumple los objetivos propuestos, y permite dar continuidad al proyecto de la empresa colaboradora *The Wise Seeker*, por lo que puede considerarse un éxito.

## **7.2 TRABAJOS FUTUROS**

A pesar de haber desarrollado un modelo totalmente funcional, aún hay mucho potencial por descubrir. Puesto que se han utilizado tecnologías muy recientes, es lógico pensar que estas pueden mejorar rápidamente ya que, por ejemplo, *ChatGPT* se encuentra todavía en una fase temprana. Por ello, se debe seguir la evolución de las tecnologías de *OpenAI* con la idea de emplear las últimas versiones y, con ello, obtener un modelo superior.

Por otra parte, el sistema diseñado debe implementarse dentro de la plataforma de *The Wise Seeker*. Tras haber realizado una prueba sobre un cierto conocimiento, un usuario debe recibir una recomendación de cursos online que traten el mismo tema. Aunque el modelo en sí está acabado, queda pendiente conectarlo dentro de la plataforma. Para ello, se introducirá el sistema dentro de un microservicio que recibirá llamadas desde el *front-end*, a las que responderá con la salida del modelo.

Finalmente, existen algunas mejoras que podrían añadirse al modelo para hacerlo aún más completo. Por ejemplo, se podría incorporar una funcionalidad de *feedback* que permitiese analizar qué recomendaciones resultan ser interesantes para el usuario. Este sistema de realimentación podría guardar los cursos recomendados a los que los usuarios acceden, y aquellos que no son tan atractivos. De esta manera, se podría ajustar el modelo para filtrar aquellas recomendaciones que no llamen tanto la atención. Otro ejemplo de mejora sería la de personalizar más la recomendación de cursos para cada usuario. Por ejemplo, se podría detectar qué preguntas ha fallado en la prueba y hacer que los cursos recomendados estén más enfocados a esas cuestiones en particular.

En conclusión, como se puede ver, el modelo diseñado tiene más potencial, por lo que se realizarán trabajos futuros para aprovecharlo.



## Capítulo 8. BIBLIOGRAFÍA

- [1] N. Seth, «Analytics Vidhya,» 28 Junio 2021. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/>.
- [2] A. Y. N. M. I. J. David M. Blei, «Latent Dirichlet Allocation,» 3 Enero 2003. [En línea]. Available: <https://web.archive.org/web/20120501152722/http://jmlr.csail.mit.edu/papers/v3/blei03a.html>.
- [3] S. Kapadia, «Towards Data Science,» 15 Abril 2015. [En línea]. Available: <https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0>.
- [4] Spacy, «Embeddings, Transformers and Transfer Learning,» [En línea]. Available: <https://spacy.io/usage/embeddings-transformers>.
- [5] Google Cloud Natural Language AI, «Demostración de la API de Natural Language,» [En línea]. Available: <https://cloud.google.com/natural-language?hl=es>.
- [6] Amazon Web Services, «What is Amazon Comprehend?,» [En línea]. Available: <https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html>.
- [7] IBM, «About IBM Watson Natural Language Understanding,» [En línea]. Available: <https://cloud.ibm.com/docs/natural-language-understanding?topic=natural-language-understanding-about>.

- [8] IBM Watson Natural Language Understanding, «IBM Watson NLU Text Analysis,» [En línea]. Available: <https://www.ibm.com/demos/live/natural-language-understanding/self-service/home>.
- [9] N. Reimers, «OpenAI GPT-3 Text Embeddings - Really a new state-of-the-art in dense text embeddings?,» 28 Enero 2022. [En línea]. Available: [https://medium.com/@nils\\_reimers/openai-gpt-3-text-embeddings-really-a-new-state-of-the-art-in-dense-text-embeddings-6571fe3ec9d9](https://medium.com/@nils_reimers/openai-gpt-3-text-embeddings-really-a-new-state-of-the-art-in-dense-text-embeddings-6571fe3ec9d9).
- [10] OpenAI, «Embeddings - OpenAI API,» [En línea]. Available: <https://platform.openai.com/docs/guides/embeddings>.
- [11] OpenAI, «Chat completion - OpenAI API,» [En línea]. Available: <https://platform.openai.com/docs/guides/chat>.
- [12] Google Cloud Natural Language, «Categorías de contenido,» [En línea]. Available: <https://cloud.google.com/natural-language/docs/categories?hl=es-419>.
- [14] Microsoft Azure, «What is Azure Cognitive Service for Language?,» [En línea]. Available: <https://learn.microsoft.com/es-es/azure/cognitive-services/language-service/overview>.

# **ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS**

Este proyecto se alinea principalmente con dos de los objetivos de desarrollo sostenible.

En primer lugar, este trabajo busca hacer una contribución en un campo que todavía está en una fase temprana. Con el desarrollo de un programa de análisis de procesamiento del lenguaje natural, se añade una nueva solución al mundo de la inteligencia artificial que, además, busca mejorar las que ya existen. Por tanto, fomenta la innovación tecnológica y se alinea con el ODS de industria, innovación e infraestructura.

En segundo lugar, el estudio y desarrollo de las técnicas de *NLP* busca, al fin y al cabo, automatizar procesos que son tediosos y poco eficientes si son realizados por un humano. Por ello, se persigue optimizar el rendimiento de los puestos de trabajo, mejorando las herramientas disponibles, y, en última instancia, fomentando un crecimiento económico. Por tanto, el proyecto se alinea con el ODS de trabajo decente y crecimiento económico.

## **ANEXO II: REPOSITORIO EN GITHUB**

Como se menciona en el capítulo de descripción de tecnologías (*Capítulo 2.* ), uno de los objetivos era dejar subido en GitHub un repositorio público el modelo final. Esto se ha hecho, y el repositorio puede accederse mediante el siguiente enlace: <https://github.com/NCSanto01/NLP-OpenAI>

En este repositorio, vienen todas las instrucciones para ejecutar el modelo desde el principio, partiendo desde cero, y con todos los pasos detallados en un archivo *README.md*. En el repositorio, se incluye un archivo para generar los textos objetivo con la estructura correcta con algunos textos de ejemplo, otro archivo para generar la caracterización de los textos objetivo, así como los *embeddings* y otro archivo para ejecutar el modelo.

Además, se incluye también una pequeña aplicación hecha con *Streamlit* con la que se puede ejecutar el modelo de manera más cómoda, visual e intuitiva. En la *Figura 30* se muestra esta aplicación.

First, define the input element, or choose from one of the source elements

## Select input element from source elements

The following elements represent online courses:

Select an element

Databases

## Or create your own

Create your own element

Input element:

```
{
  "id" : "100"
  "name" : "Databases"
  "content" :
  "Learn database development from our best-in-class instructors. Our courses help you skill up in SQL, Python, NoSQL, Object relational mapping and more. Our authors show you how to gather the information needed, analyze the requirements, design a schema and implement the final solution."
}
```

## Define ChatGPT prompt

Here you must define the prompt that ChatGPT will use to generate the keywords for your input element. You can use the default prompt only if the input element represents an online course.

Type the prompt

This is the information about an online course. You must generate 5 words that describe as precisely as

## Generate recommendations

Get recommended source elements based on the input element and prompt

Max Distance



Selected Max Distance: 0.11

Generate recommendations

## Results

Input Element name: Databases

Keywords: Database Development, SQL, Python, NoSQL, Object-relational mapping.

Recommendations:

	course	distance
0	Databases	0.037

*Figura 30. Página de ejecución del modelo*