GRADO EN ANÁLISIS DE NEGOCIOS

TRABAJO FIN DE GRADO

Dashboard interactivo para el análisis organizacional en la comunicación empresarial

Autor
Juan Carlos Vecino de Haro

Dirigido por
Javier Matanza Domingo

Madrid
Abril 2025

GRADO EN ANÁLISIS DE NEGOCIOS

TRABAJO FIN DE GRADO

Dashboard interactivo para el análisis organizacional en la comunicación empresarial

Autor
Juan Carlos Vecino de Haro

Dirigido por
Javier Matanza Domingo

Madrid
Abril 2025

Resumen

Palabras Clave

Procesamiento de Lenguaje Natural (PLN); Comunicación Organizacional; Panel Interactivo; Análisis de Emociones; Análisis de Redes Sociales; Conjunto de Datos de Correos de Enron

0.1. Resumen ejecutivo

Este Trabajo Fin de Grado, titulado "Dashboard interactivo para el análisis organizacional en la comunicación empresarial", presenta el desarrollo de una herramienta visual interactiva para analizar la comunicación interna en organizaciones a través de técnicas avanzadas de procesamiento de lenguaje natural (NLP) y visualización de datos. El caso de estudio se basa en el Enron Email Dataset, una base de datos pública que contiene más de 500.000 correos electrónicos reales entre empleados de la empresa Enron, lo que permite explorar dinámicas comunicativas auténticas en un entorno corporativo de gran escala.

El sistema desarrollado permite realizar un análisis profundo de las interacciones internas mediante tres funcionalidades principales: detección de emociones y sentimientos en los correos, análisis de la red de comunicación entre empleados, y visualización temporal y temática de los mensajes. Estas funcionalidades permiten identificar patrones relevantes para el diagnóstico organizacional, como posibles cuellos de botella en la comunicación, niveles de cohesión entre departamentos, o señales tempranas de conflicto o desmotivación.

El dashboard ha sido diseñado con un enfoque intuitivo, facilitando su uso por parte de perfiles no técnicos, como responsables de recursos humanos o consultores organizativos. Además, se incluyen propuestas de mejora y expansión futura del sistema, como la integración de otros datasets, la incorporación de análisis multilingüe o la conexión con herramientas empresariales reales.

En conjunto, este proyecto demuestra el potencial del análisis de datos textuales aplicado al ámbito empresarial, ofreciendo una herramienta práctica para mejorar la comprensión de las dinámicas internas y apoyar la toma de decisiones estratégicas basadas en evidencia.

0.2. Introducción

La transformación digital ha impulsado la adopción de tecnologías en el ámbito corporativo, pero entender las relaciones humanas sigue siendo clave para mejorar la eficiencia

y el bienestar en las organizaciones. Herramientas basadas en IA y NLP, como las desarrolladas por empresas como Lattice o SuccessKPI, permiten analizar grandes volúmenes de datos textuales con el fin de detectar patrones ocultos en la comunicación interna. En este contexto, este trabajo se propone aplicar dichas tecnologías al análisis de correos corporativos para generar inteligencia organizacional.

0.3. Definición del proyecto

El objetivo de este proyecto es desarrollar un dashboard interactivo que permita analizar de forma automática las comunicaciones internas de una organización mediante técnicas avanzadas de procesamiento de lenguaje natural (NLP) y análisis de redes. El sistema se enfocará en la extracción de sentimientos, emociones y relaciones entre empleados a partir de correos electrónicos, sin requerir intervención manual constante ni etiquetado adicional.

El dashboard integrará modelos preentrenados de análisis emocional como roberta-base-go_emotions, junto con visualizaciones dinámicas creadas con Dash y Plotly. Además, se diseñará un grafo interactivo para representar las conexiones comunicativas, permitiendo identificar estructuras informales, cuellos de botella y actores clave dentro de la organización.

La meta es proporcionar una herramienta accesible, escalable y basada en datos que permita a los equipos directivos detectar dinámicas organizativas, mejorar la gestión del talento y optimizar la estructura interna de la empresa mediante una toma de decisiones informada y objetiva.

0.4. Dashboard

El dashboard desarrollado cuenta con tres secciones principales:

- Overview, que presenta KPIs, evolución temporal, y distribución de emociones;
- Network Analysis, que visualiza la red de comunicaciones mediante grafos y métricas como densidad y longitud media de caminos;
- Content Analysis, que explora temas y palabras clave, con su respectivo análisis emocional y evolución en el tiempo.

Para comprobar la utilidad de la herramienta, se ha utilizado como ejemplo el dataset de correos electrónicos de la organización Enron, dando lugar al siguiente dashboard:

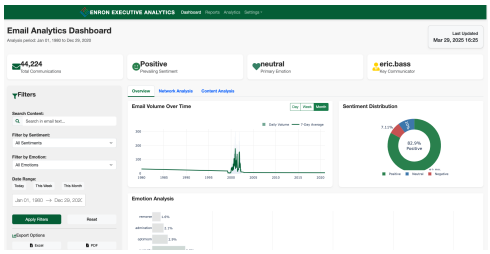

Figura 1: Vista general del dashboard (1)

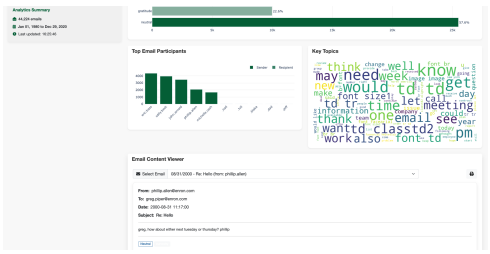

Figura 2: Vista general del dashboard (2)

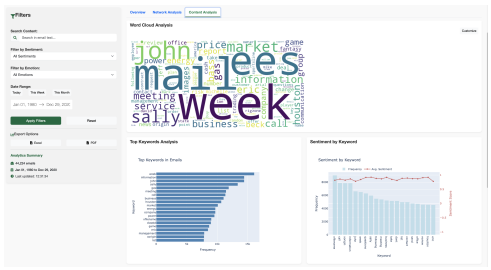

Figura 3: Análisis de contenido (1)

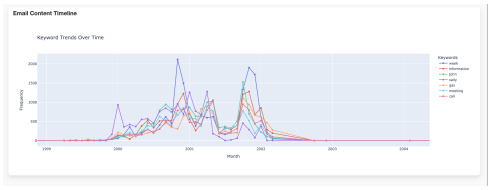

Figura 4: Análisis de contenido (2)

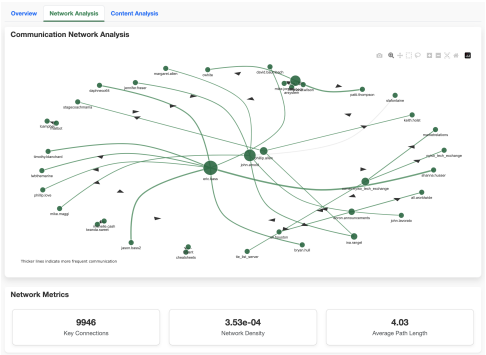

Figura 5: Red de comunicaciones

Figura 6: Capturas del dashboard interactivo desarrollado

Estas funcionalidades permiten a los usuarios identificar cuellos de botella, detectar liderazgos informales y analizar el tono emocional de las comunicaciones, todo desde una interfaz accesible e interactiva.

0.5. Resultados y Conclusiones

El análisis del caso Enron ha revelado hallazgos relevantes, como una sorprendente predominancia de sentimientos positivos o neutros en un contexto de crisis, lo que podría reflejar mecanismos de autocensura o contención emocional. El grafo de comunicaciones mostró una red jerárquica con baja densidad y alta dependencia de nodos clave, sugiriendo vulnerabilidades estructurales. La herramienta desarrollada no solo facilita el análisis retrospectivo, sino que puede actuar como un sistema de alerta temprana y una base objetiva para decisiones estratégicas. Entre las líneas futuras se contempla la integración de modelos personalizados, sistemas de búsqueda semántica (RAG), y mejoras en la clasifi-

cación de emociones, consolidando este trabajo como una solución escalable e innovadora para el análisis de la comunicación empresarial.

0.6. Referencias

[1] Mordor Intelligence, *Análisis de participación y tamaño del mercado de tecnología de Big Data: tendencias y pronósticos de crecimiento (2024-2029),* Mordor Intelligence, Informe de investigación de mercado, 2024. Disponible en: `https://www.mordorintelligence.com/es/industry-reports/fdi-perspective-of-big-data-technology`.

[2] E. Covas, *Named Entity Recognition using GPT for Identifying Comparable Companies,* 2023. arXiv:2307.07420 [cs.CL]. Disponible en: `https://arxiv.org/abs/2307.07420`.

[3] W. W. Cohen, *Enron Email Dataset,* Carnegie Mellon University, 2015. Disponible en: `https://www.cs.cmu.edu/~enron/`.

Proyect Summary

Keywords

Natural Language Processing (NLP); Organizational Communication; Interactive Dashboard; Emotion Analysis; Social Network Analysis; Enron Email Dataset

0.7. Abstract

This Final Degree Project, titled "Interactive Dashboard for Organizational Analysis in Business Communication", presents the development of an interactive visual tool designed to analyze internal communication within organizations using advanced Natural Language Processing (NLP) techniques and data visualization. The case study focuses on the Enron Email Dataset, a public collection of over 500,000 real emails exchanged between Enron employees, enabling the exploration of authentic communicative dynamics in a large corporate environment.

The implemented system enables in-depth analysis of internal interactions through three main features: emotion and sentiment detection in emails, communication network analysis between employees, and temporal and thematic visualization of messages. These functionalities allow for the identification of relevant patterns for organizational diagnosis, such as communication bottlenecks, departmental cohesion levels, or early signs of conflict or disengagement.

The dashboard has been designed with an intuitive user interface, making it accessible to non-technical profiles such as HR professionals or organizational consultants. Furthermore, potential improvements and future lines of development are proposed, including integration with additional datasets, support for multilingual analysis, and connection with real-world business tools.

Overall, this project demonstrates the potential of textual data analysis in the business domain, providing a practical tool to enhance the understanding of internal dynamics and support strategic decision-making based on evidence.

0.8. Introduction

Digital transformation has driven the adoption of technologies in the corporate environment, but understanding human relationships is still key to improving efficiency and well-being in organizations. Tools based on AI and NLP, such as those developed by companies like Lattice or SuccessKPI, allow analyzing large volumes of textual data in order to detect hidden patterns in internal communication. In this context, this paper proposes

to apply these technologies to the analysis of corporate emails to generate organizational intelligence.

0.9. Project definition

The objective of this project is to develop an interactive dashboard to automatically analyze an organization's internal communications using advanced natural language processing (NLP) and network analysis techniques. The system will focus on extracting sentiments, emotions and relationships between employees from emails, without requiring constant manual intervention or additional labeling.

The dashboard will integrate pre-trained models of emotional analysis such as roberta-base-goo_emotions, along with dynamic visualizations created with Dash and Plotly. In addition, an interactive graph will be designed to represent the communicative connections, allowing the identification of informal structures, bottlenecks and key players within the organization.

The goal is to provide an accessible, scalable and data-driven tool that allows management teams to detect organizational dynamics, improve talent management and optimize the internal structure of the company through informed and objective decision making.

0.10. Dashboard

The developed dashboard has three main sections:

- Overview, which presents KPIs, time evolution, and distribution of emotions;
- Network Analysis, which visualizes the communications network through graphs and metrics such as density and average path length;
- Content Analysis, which explores topics and keywords, with their respective emotional analysis and evolution over time.

To test the usefulness of the tool, the e-mail dataset of the Enron organization was used as an example, resulting in the following dashboard:

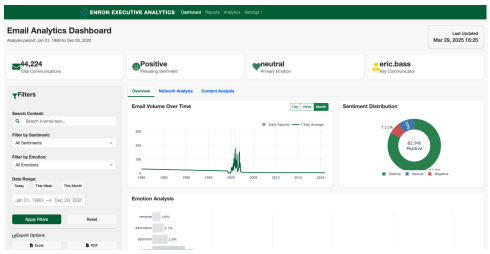

Figura 7: Dashboard overview (1)

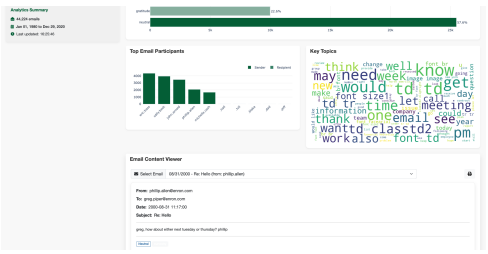

Figura 8: Dashboard overview (2)

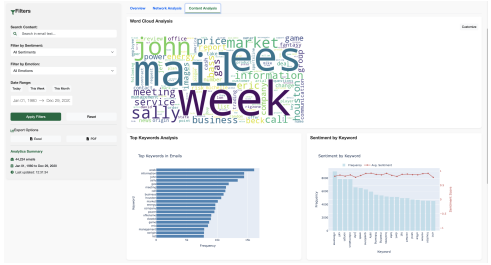

Figura 9: Content analysis (1)

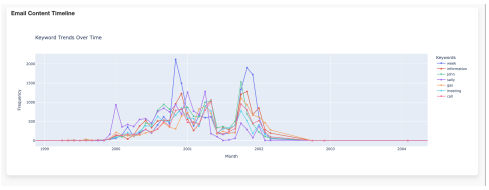

Figura 10: Content analysis (2)

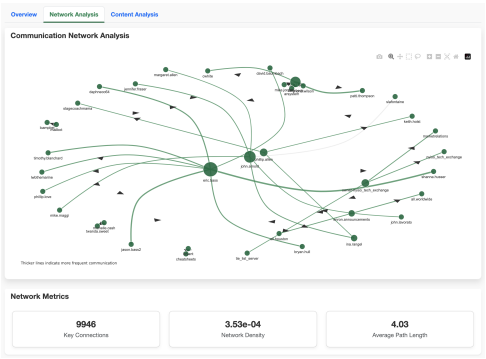

Figura 11: Communications network

Figura 12: Screenshots of the developed interactive dashboard

These functionalities enable users to identify communication bottlenecks, detect informal leadership structures, and analyze the emotional tone of exchanges — all through an accessible and interactive interface.

0.11. Results and Conclusions

The analysis of the Enron case has revealed relevant findings, such as a surprising predominance of positive or neutral sentiments in a crisis context, which may reflect mechanisms of self-censorship or emotional restraint. The communications graph displayed a hierarchical network with low density and high dependency on key nodes, suggesting structural vulnerabilities.

The developed tool not only facilitates retrospective analysis but can also serve as an early warning system and an objective basis for strategic decision-making. Future work includes the integration of customized models, semantic search systems (RAG),

and improvements in emotion classification — positioning this project as a scalable and innovative solution for business communication analysis.

0.12. References

[1] Mordor Intelligence, *Big Data Technology Market Share Analysis and Size: Trends and Growth Forecasts (2024–2029),* Mordor Intelligence, Market Research Report, 2024. Available at: `https://www.mordorintelligence.com/es/industry-reports/fdi-perspective-of-big-data-technology`.

[2] E. Covas, *Named Entity Recognition using GPT for Identifying Comparable Companies,* 2023. arXiv:2307.07420 [cs.CL]. Available at: `https://arxiv.org/abs/2307.07420`.

[3] W. W. Cohen, *Enron Email Dataset,* Carnegie Mellon University, 2015. Available at: `https://www.cs.cmu.edu/~enron/`.

Índice general

Índice de figuras

Índice de cuadros

Listings

Acrónimos

NLP	Natural language processing
LLM	Large Language Model

Capítulo 1

Introducción

1.1. Motivación

En los últimos años, especialmente a raíz de la pandemia de COVID-19, la adopción de tecnologías en los entornos laborales ha experimentado un notable incremento. Sin embargo, comprender las relaciones humanas dentro de una organización sigue siendo esencial para potenciar la eficacia, la cooperación y el bienestar de los empleados. El análisis de las interacciones laborales ha cobrado una relevancia significativa, ya que proporciona información valiosa sobre la estructura interna, la comunicación entre equipos y la cultura corporativa.

Empresas como Lattice y ChartHop se dedican a analizar los datos internos de las organizaciones para optimizar la gestión del talento y mejorar la comunicación interna. Estas herramientas permiten a las empresas visualizar su estructura organizativa y facilitar la toma de decisiones estratégicas basadas en datos precisos.

La inversión en soluciones de análisis de datos empresariales ha crecido exponencialmente en los últimos años. Según datos de Statista, el mercado global de analítica de datos alcanzó los 234.300 millones de dólares en 2024[1], lo que refleja la importancia que las empresas otorgan a la interpretación y utilización efectiva de sus datos internos.

El Procesamiento de Lenguaje Natural (NLP) y la Inteligencia Artificial (IA) han emergido como herramientas fundamentales en el análisis empresarial. Estas tecnologías permiten a las máquinas comprender y procesar el lenguaje humano, facilitando la automatización de tareas y el análisis de grandes volúmenes de datos textuales. Por ejemplo, empresas como SucessKPI utilizan NLP para analizar comentarios de clientes, correos electrónicos y otras formas de comunicación interna, identificando patrones y áreas de mejora.

Diversos estudios han explorado la aplicación de modelos de NLP en el ámbito empresarial. Por ejemplo, el uso de modelos de lenguaje de gran tamaño, como GPT, ha demostrado una alta precisión en la identificación de empresas comparables a partir de descripciones textuales, lo que es crucial para la valoración de empresas en el sector de capital privado [2]. Además, se ha desarrollado un marco de aprendizaje profundo para medir la estrategia digital de las empresas analizando las transcripciones de las conferencias de los informes de resultados, lo que ayuda a comprender los diferentes patrones de estrategia digital adoptados [3], llegando a segmentar empresas según sus transcripciones como se muestra en la figura 1.1:

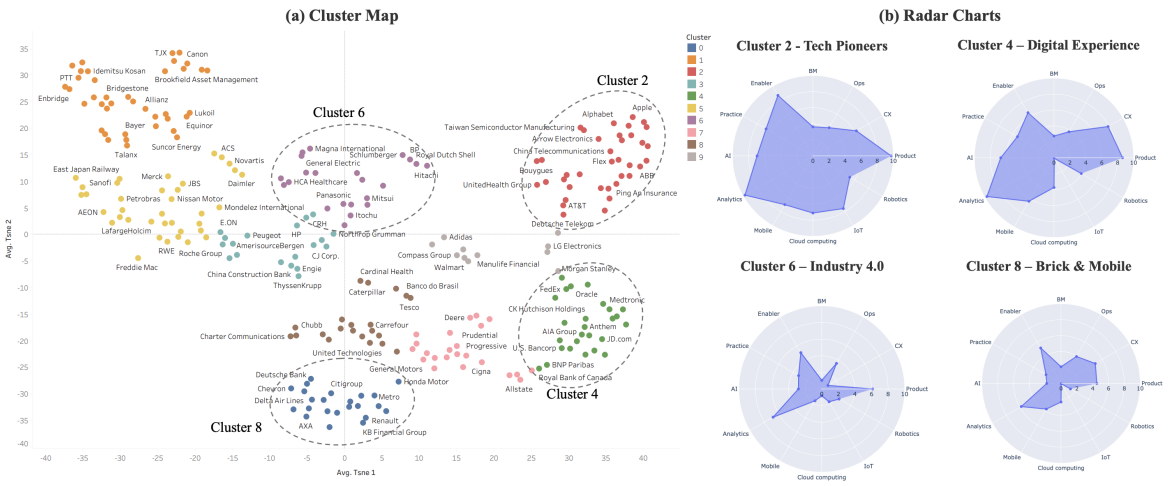

Figura 1.1: Cluster basado en las earnings calls de empresas

La implementación de estas tecnologías permite identificar cuellos de botella, deficiencias en la comunicación y oportunidades de mejora que anteriormente requerían un coste muy elevado para detectarlas. Al analizar las interacciones dentro de una empresa, es posible optimizar procesos, mejorar la colaboración entre equipos y, en última instancia, aumentar la productividad y satisfacción de los empleados.

1.2. Objetivo

El principal objetivo de este trabajo es el desarrollo de un dashboard interactivo que permita visualizar, de manera rápida y clara, las comunicaciones internas dentro de una empresa. A través de esta herramienta, se podrán observar quién se comunica con quién, los temas tratados, el sentimiento expresado, entre otros aspectos clave. Esto permitirá detectar ineficiencias dentro de la estructura organizativa y mejorar la toma de decisiones en la gestión del talento.

Para alcanzar este objetivo, se desarrollará un modelo automatizado que evalúe la comunicación dentro de los departamentos de una empresa. Se analizarán las conversaciones entre los miembros de un mismo departamento (a través de correos electrónicos), con el fin de identificar patrones de comunicación, colaboración y posibles ineficiencias organizativas.

A partir de este análisis, el proyecto busca diseñar y desarrollar una herramienta gráfica basada en un modelo SaaS (Software as a Service), que facilite la visualización intuitiva de las relaciones entre empleados y departamentos. Esta herramienta permitirá identificar problemas de comunicación, estructuras informales y oportunidades de mejora dentro de la organización. Su propósito es proporcionar una base objetiva para la toma de decisiones estratégicas en la gestión del talento y la optimización de la estructura organizativa.

En línea con los avances tecnológicos en analítica de datos y ciencia de redes, este trabajo representa una innovación en el ámbito de la gestión de recursos humanos. Mediante la aplicación de metodologías avanzadas de análisis de grafos y procesamiento de lenguaje natural (NLP), se busca superar las limitaciones de los enfoques tradicionales, como encuestas y evaluaciones subjetivas, proporcionando una herramienta capaz de obtener información en tiempo real sobre la dinámica interna de la empresa.

Capítulo 2

Marco Teórico y estado del arte

2.1. Embedding

A lo largo de este trabajo se analizarán correos electrónicos, sin embargo, una red neuronal no puede interpretar texto en su forma original. Es necesario transformar las frases en vectores numéricos para que puedan ser procesadas por modelos de aprendizaje automático. Esta transformación recibe el nombre de tokenización.

Una de las estrategias más básicas para esta tarea es el *One-hot encoding*, que consiste en representar cada término del vocabulario mediante un vector, en el que solo una posición toma el valor 1, mientras que el resto son ceros. Cada dimensión del vector corresponde a una palabra única en el vocabulario. Por ejemplo, si el vocabulario está compuesto por las palabras 'gas', 'trading' y 'rueda', las representaciones serían las siguientes:

$$gas = [1, 0, 0]$$

$$trading = [0, 1, 0]$$

$$rueda = [0, 0, 1]$$

Esta representación, aunque sencilla, presenta limitaciones considerables: no refleja relaciones semánticas entre términos y se vuelve ineficiente cuando el tamaño del vocabulario crece. Además, la distancia entre vectores es uniforme, lo que impide distinguir similitudes entre conceptos. Si se calcula la distancia euclidiana entre dos vectores $\mathbf{p}$ y $\mathbf{q}$, se tiene:

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2.1}$$

Como todos los vectores son ortogonales[1], esta distancia es idéntica entre cualquier par de palabras. Por lo tanto, este modelo nos indica que la relación semántica entre *trading* y *gas* es la misma que *trading* y *rueda*.

Para superar estas limitaciones, Google propuso el modelo *Word2Vec*[10], una técnica de representación vectorial de palabras que proyecta cada término en un espacio de baja dimensión (usualmente entre 100 y 300), permitiendo capturar tanto relaciones semánticas como sintácticas.

[1] Forman un ángulo recto

Word2Vec puede entrenarse mediante dos arquitecturas principales: CBOW (Continuous Bag of Words) y Skip-gram. En esta última, el objetivo del modelo es predecir las palabras que rodean a una palabra central. Por ejemplo, dada la palabra 'trading', el modelo intentaría predecir términos como 'Forex', 'long' o 'derivatives'.

Este tipo de modelos suele entrenarse usando variantes optimizadas del algoritmo Softmax, como el Softmax Jerárquico. Este método organiza el vocabulario en un árbol binario, a menudo construido con codificación de Huffman. Cada palabra se ubica en una hoja del árbol, y la probabilidad de una palabra se calcula como el producto de las probabilidades a lo largo del camino desde la raíz hasta dicha hoja:

$$P(w) = \prod_{n \in \text{path to } w} P(n) \tag{2.2}$$

donde n_k representa los nodos intermedios en la ruta hacia la palabra w, y $P(n_k)$ es la probabilidad de elegir correctamente el siguiente nodo en cada bifurcación.

En el caso de tener un vocabulario con palabras relacionadas con el sector bancario, como 'crédito', 'depósito', 'interés', 'hipoteca' y 'cliente', en lugar de calcular individualmente la probabilidad de cada término comparándolo contra todas las demás palabras del vocabulario, el modelo recurre a una estructura jerárquica en forma de árbol binario.

Este árbol permite clasificar progresivamente las palabras mediante decisiones binarias. Por ejemplo, se podría empezar diferenciando entre conceptos financieros y no financieros. A continuación, dentro del grupo de términos financieros, el modelo distinguiría entre productos (como 'hipoteca' o 'crédito') y actores (como 'cliente'). Finalmente, seguiría ramificando hasta alcanzar la palabra objetivo, como 'hipoteca'.

Este mecanismo ofrece una forma eficiente de cálculo, ya que evita recorrer el vocabulario completo. Solo se evalúan las decisiones que conducen a una palabra específica, reduciendo la complejidad computacional de $O(N)$ a $O(\log N)$, siendo N el tamaño del vocabulario.

Durante el proceso de entrenamiento, el modelo ajusta los vectores de las palabras utilizando técnicas de optimización como el descenso de gradiente estocástico. En el modelo *Skip-gram*, el objetivo es maximizar la probabilidad condicional de observar las palabras del contexto w_c dado un término central w_t, a través de la siguiente función:

$$p(w_c \mid w_t) = \frac{\exp(\vec{w_t} \cdot \vec{w_c})}{\sum_{w \in V} \exp(\vec{w_t} \cdot \vec{w})} \tag{2.3}$$

Aquí, $\vec{w}$ representa el vector asociado a la palabra w, y el producto escalar entre vectores indica su nivel de similitud. La función objetivo busca asignar mayores probabilidades a las palabras reales del contexto, ajustando los vectores para que las palabras relacionadas estén más próximas en el espacio vectorial.

Como resultado del entrenamiento, se obtienen vectores que agrupan palabras con significados similares en regiones próximas del espacio. Las dimensiones de estos vectores dejan de tener un significado individual claro (como ocurría con el One-hot encoding), y pasan a representar atributos abstractos, como 'nacionalidad' o 'naturaleza emocional'. A diferencia del caso anterior, ahora la distancia entre vectores refleja la similitud entre conceptos. Por ejemplo, en la Figura 2.2 se muestra una visualización de los términos más cercanos a 'Spain' en un modelo Word2Vec entrenado con 10.000 palabras:

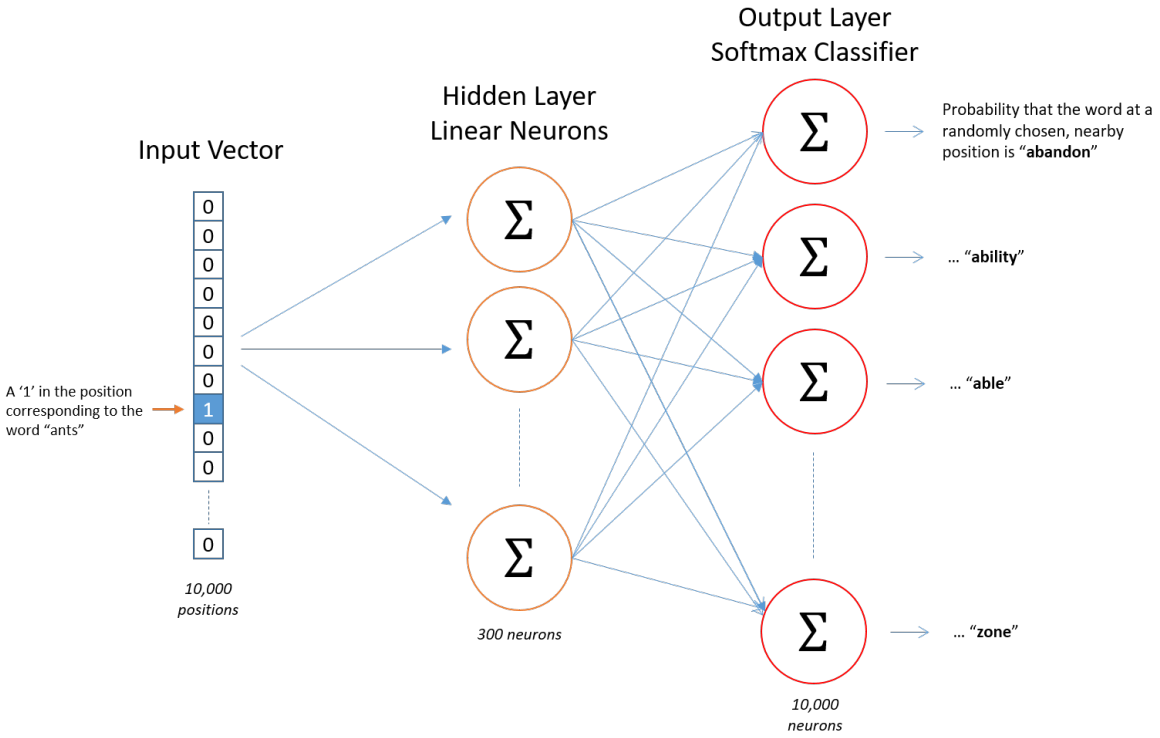

Figura 2.1: Arquitectura Skip-gram

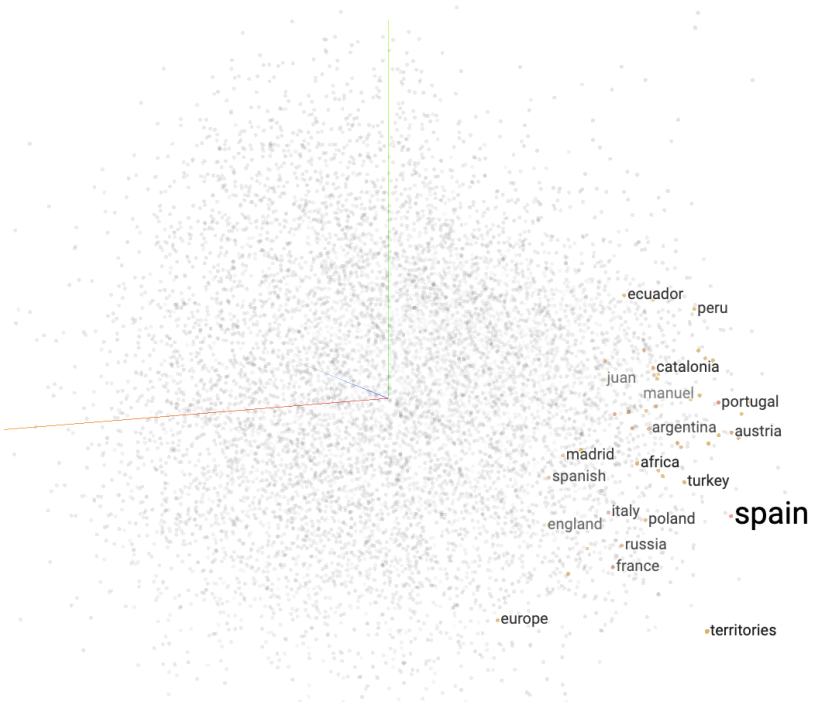

Figura 2.2: Ejemplo de Word2Vec con la palabra 'Spain'

Se puede observar que términos como 'Portugal', 'France' o 'Spanish' aparecen entre los más próximos a 'Spain', demostrando la capacidad del modelo para capturar relaciones semánticas relevantes.

En conclusión, los *embeddings* permiten representar texto en vectores densos y conti-

nuos, reduciendo la dimensionalidad del problema sin perder capacidad expresiva, lo que mejora significativamente la comprensión del lenguaje por parte de los modelos.

2.1.1. Embeddings Posicionales

Los modelos basados únicamente en embeddings estáticos presentan dos carencias fundamentales: no consideran la posición de las palabras dentro de la secuencia, ni ajustan su representación según el contexto de las demás palabras. Para solventar estas limitaciones, los modelos Transformer introducen el mecanismo de atención y la codificación posicional.

Supongamos la frase:

"El médico prescribió medicamentos porque el paciente tenía la fiebre"

En este contexto, sería esperable anticipar que la siguiente palabra podría ser *"alta"*. Para lograr dicha predicción, es esencial identificar qué palabras ofrecen mayor valor informativo, como *"médico"*, *"prescribió"*, *"paciente"* y *"fiebre"*, y al mismo tiempo reducir el peso de términos funcionales o menos relevantes como *"el"* o *"porque"*. El mecanismo de atención permite realizar esta distinción ponderando la relevancia de cada palabra dentro del conjunto de entrada.

El mecanismo de atención comienza al seleccionar una palabra específica dentro de la secuencia, que se utilizará como punto de referencia. Supongamos que dicha palabra es *"fiebre"*. Esta se transforma en una *query* (Q), la cual se utilizará para evaluar qué otras palabras del contexto aportan información útil. La generación de la query se realiza aplicando una transformación lineal al embedding de la palabra elegida:

$$Q = X_{\text{fiebre}} W^Q \tag{2.4}$$

En esta fórmula, X_{fiebre} representa el vector de embedding de la palabra seleccionada y W^Q es una matriz de pesos entrenable que proyecta dicho embedding en el espacio de consultas.

En paralelo, los embeddings de cada palabra dentro de la secuencia son proyectados a dos nuevos espacios vectoriales para generar los denominados *keys* (K) y *values* (V). Los vectores key permiten evaluar la relevancia de una palabra en función de una consulta dada (query), mientras que los vectores value encapsulan la información que dicha palabra puede aportar al resultado final. Esta transformación se realiza aplicando matrices de pesos entrenables de la siguiente forma:

$$K = X_{\text{palabra}} W^K, \quad V = X_{\text{palabra}} W^V \tag{2.5}$$

donde X_{palabra} corresponde al embedding de una palabra cualquiera en la secuencia, y W^K y W^V son matrices de parámetros aprendidas durante el entrenamiento, independientes de la matriz utilizada para generar la query. Estas matrices tienen como objetivo reconfigurar los vectores originales para facilitar la comparación de relevancia y la agregación de contenido contextual.

Una vez obtenidos los vectores K y Q, se procede a calcular la similitud entre la consulta y cada key a través del producto escalar. Este cálculo indica qué tan relevante es cada palabra de la secuencia en relación con la palabra que actúa como foco de atención. Cuanto mayor es el producto escalar entre Q y un vector K, mayor será la atención asignada a esa palabra. Para convertir estas similitudes en una distribución de probabilidades, se aplica la función *softmax*, dando lugar a los llamados pesos de atención:

$$\text{Pesos de Atención} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{2.6}$$

El término $\sqrt{d_k}$ actúa como factor de normalización, donde d_k es la dimensión de los vectores key, con el fin de mantener los valores dentro de una escala manejable y evitar gradientes extremos. Los pesos resultantes indican la importancia relativa de cada palabra dentro del contexto, y guían la combinación ponderada de los valores V para generar una salida contextualizada.

Una vez calculados los pesos de atención, estos se utilizan para combinar los vectores V asociados a cada palabra del contexto. Específicamente, cada vector value se multiplica por su correspondiente peso de atención, y el resultado es una suma ponderada que produce la salida del mecanismo de atención:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.7}$$

Este procedimiento genera una matriz de salida donde cada fila representa una agregación contextualizada de los vectores V, determinada por la importancia relativa de cada palabra según la query. En otras palabras, la salida de una atención es un conjunto de vectores que reflejan qué partes del texto fueron consideradas más relevantes al procesar una palabra concreta.

De forma análoga a cómo una red convolucional puede detectar patrones específicos en diferentes regiones de una imagen, cada *Attention Head* en un modelo Transformer se especializa en captar distintos tipos de relaciones dentro del texto: algunas pueden centrarse en asociaciones gramaticales, otras en aspectos semánticos o dependencias a largo plazo.

Por esta razón, se utiliza el mecanismo de *Multi-Head Attention*, que consiste en aplicar varias atenciones de forma paralela. Cada cabeza aprende un patrón de atención distinto, y sus salidas se combinan posteriormente para formar una representación más rica y diversa del texto de entrada.

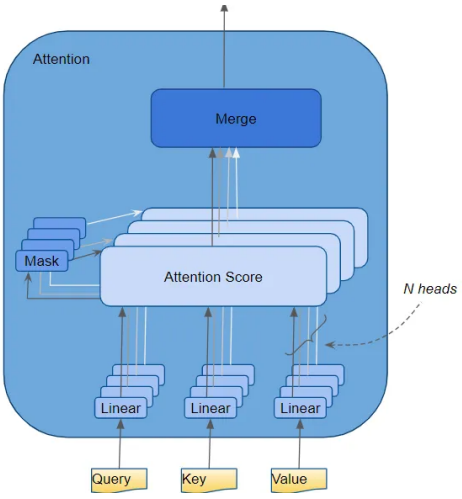

Figura 2.3: Ilustración de una capa de atención con múltiples cabezas

Es fundamental destacar que los embeddings procesados por el mecanismo de atención no solo contienen información sobre el significado de cada palabra, sino también sobre

su posición en la secuencia. Esto se logra mediante la incorporación de codificaciones posicionales, que permiten al modelo interpretar correctamente el orden de las palabras.

La salida final de cada *Attention Head*, enriquecida con esta información posicional, constituye una representación contextual que es esencial para tareas como la predicción de la siguiente palabra o la comprensión de dependencias complejas dentro del texto.

2.2. Transformers

El modelo Transformer fue introducido por Vaswani et al. en 2017 [12] como una nueva arquitectura diseñada específicamente para el procesamiento de secuencias, particularmente en tareas de lenguaje natural. Su principal innovación consiste en el uso exclusivo de mecanismos de atención, eliminando la necesidad de recurrencia, característica fundamental de modelos anteriores como las redes neuronales recurrentes (RNN), las LSTM o las GRU. Esta decisión permite un entrenamiento más eficiente, altamente paralelizable y con mejor capacidad para capturar relaciones de largo alcance entre palabras dentro de una misma secuencia.

La arquitectura estándar del Transformer está formada por dos bloques principales: un codificador (*encoder*) y un decodificador (*decoder*). El codificador se encarga de procesar la secuencia de entrada y construir una representación contextual de la misma, mientras que el decodificador utiliza esa representación para generar una salida, palabra por palabra. Este enfoque ha sido clave en tareas como la traducción automática, el resumen de texto y la generación de lenguaje natural.

Cada uno de estos bloques está compuesto por una pila de capas idénticas. En el codificador, cada capa contiene una subcapa de *Multi-Head Attention* (atención con múltiples cabezas) seguida de una red completamente conectada (feed-forward). Estas subcapas están envueltas por mecanismos de normalización y conexiones residuales que estabilizan y aceleran el entrenamiento. El decodificador añade una tercera subcapa de atención que le permite centrarse en partes específicas de la salida del codificador mientras genera el texto.

Desde su publicación, la arquitectura Transformer ha sido la base de múltiples modelos avanzados que han batido récords en distintas tareas de NLP. Algunos de los más representativos son:

- **BERT (Encoder)**: Enfocado en comprensión de texto, clasificación y análisis semántico.
- **T5 (Encoder-Decoder)**: Aplicado principalmente a traducción, resumen y reformulación de texto.
- **GPT-3 (Decoder)**: Optimizado para generación de texto coherente y creativo.

Tipo	Ejemplo	Aplicación
Codificador	BERT	Clasificación de texto
Codificador-Decodificador	T5	Traducción y tareas multitarea
Decodificador	GPT-3	Generación de texto

Cuadro 2.1: Tipos de modelos derivados del Transformer

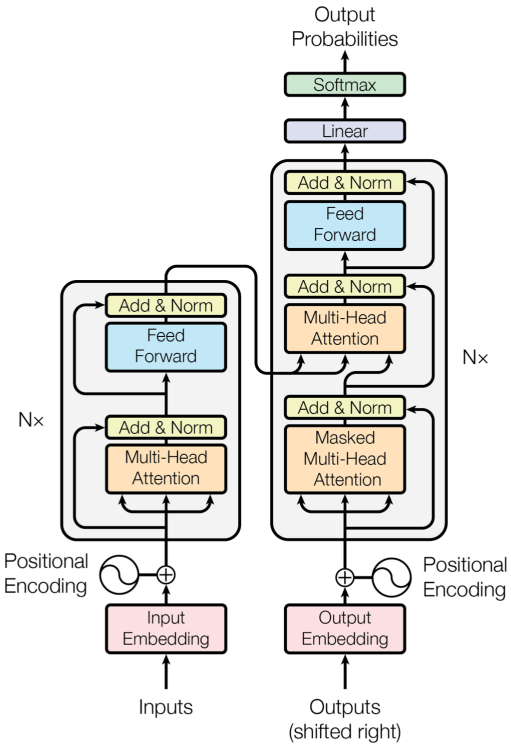

Figura 2.4: Arquitectura general de un Transformer Encoder-Decoder

Sin embargo, en el presente trabajo, nos centraremos exclusivamente en el uso de BERT (un modelo basado únicamente en la parte codificadora de la arquitectura Transformer) debido a que nuestro objetivo principal es realizar un análisis de sentimientos en correos electrónicos, una tarea de clasificación de texto.

Por esta razón, no se considera necesario profundizar en la arquitectura completa del Transformer ni en variantes orientadas a generación de lenguaje como GPT, ya que no son relevantes en el marco de este proyecto. Nos limitaremos a explicar los aspectos del codificador que resultan pertinentes para comprender el funcionamiento de BERT y su aplicación práctica.

2.2.1. BERT

Bidirectional Encoder Representations from Transformers (BERT) es un modelo de lenguaje basado en la arquitectura Transformer desarrollado por Google [13]. Su principal innovación radica en su capacidad de entender el contexto de una palabra en relación con su contexto completo, gracias a su entrenamiento bidireccional. Esto lo distingue de modelos anteriores como Word2Vec o GloVe, que generaban representaciones estáticas para cada palabra, sin considerar el significado en función del contexto circundante.

BERT se entrena utilizando el método de *Masked Language Model* (MLM), donde algunas palabras de una oración se ocultan aleatoriamente y el modelo debe predecirlas basándose en su contexto. Este enfoque permite que el modelo capture una comprensión profunda del significado de las palabras en distintas posiciones dentro de una frase.

2.2.2. BERT para Análisis de Sentimiento

En este proyecto, se empleará un modelo basado en BERT para la tarea de análisis de sentimiento de correos electrónicos. En particular, se utilizará el modelo preentrenado *roberta-base-go_ emotions-onnx* [14], que es una versión optimizada de RoBERTa adaptada específicamente para la clasificación de emociones en texto.

El flujo de procesamiento para la clasificación de sentimiento es el siguiente:

1. **Tokenización:** El texto del correo se convierte en una secuencia de tokens mediante un tokenizador basado en el modelo preentrenado de RoBERTa. Cada palabra o subpalabra es transformada en un identificador numérico.

2. **Codificación con BERT:** Los tokens son ingresados al modelo BERT, que genera representaciones vectoriales para cada uno, capturando la semántica contextual de la oración.

3. **Clasificación:** La representación del token especial [CLS] en la última capa del modelo se pasa a través de una capa lineal con función softmax para predecir la probabilidad de cada clase de sentimiento.

$$P(y \mid X) = \text{softmax}(W \cdot h_{[CLS]} + b) \tag{2.8}$$

   donde $h_{[CLS]}$ es la representación del token de clasificación, W es una matriz de pesos aprendida y b es un vector de sesgo.

4. **Predicción:** Se selecciona la emoción con mayor probabilidad como la clasificación final del sentimiento del correo.

Este modelo permite identificar emociones como alegría, tristeza, enfado y sorpresa en los correos electrónicos, proporcionando un análisis más detallado sobre el tono y la intención de los mensajes enviados y recibidos en la organización.

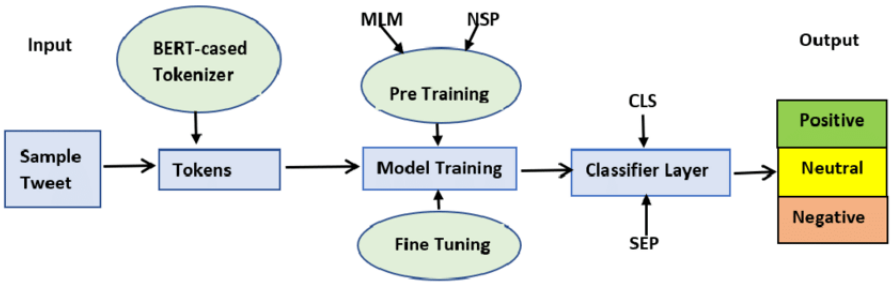

Figura 2.5: Ejemplo de análisis de sentimiento con BERT

El uso de este modelo en el presente trabajo permitirá evaluar la carga emocional de los correos electrónicos analizados, proporcionando información clave para entender dinámicas comunicativas en entornos empresariales.

2.3. Estado del Arte

El paper de Qi Wen [4], titulado *Finding top performers through email patterns analysis*, explora cómo el desempeño laboral de un empleado está estrechamente relacionado con sus estrategias de comunicación digital. Utilizando una combinación de análisis de redes sociales y análisis semántico, los autores desarrollaron un método para identificar a los empleados con mejor rendimiento a partir del análisis de sus correos electrónicos.

Para evaluar la capacidad predictiva de estos indicadores, el estudio utilizó modelos de machine learning basados en AdaBoost, logrando una precisión del *83.56 %* en la identificación de empleados de alto desempeño. Además, a través de un análisis de clusters, se identificaron tres tipos de trabajadores sobresalientes: networkers (con posiciones centrales en la red), influencers (con ideas influyentes en la organización) y positivists (con un lenguaje positivo en sus comunicaciones).

Este estudio abre nuevas oportunidades para la gestión del talento dentro de las empresas, permitiendo identificar empleados clave y optimizar la comunicación organizacional.

Otros estudios han abordado el análisis de la comunicación interna en entornos corporativos desde distintas perspectivas. Por ejemplo, el paper de Fernanda B. Viégas [5], titulado *Visualizing email content: portraying relationships from conversational histories*, propone un algoritmo capaz de extraer los temas de conversación en distintos correos electrónicos y generar una interfaz gráfica que representa la evolución de los temas tratados entre individuos a lo largo del tiempo. Esta visualización facilita la identificación de patrones de comunicación, la detección de cambios en la dinámica organizacional y la comprensión de cómo se desarrollan las relaciones laborales dentro de una empresa.

Por otro lado, el paper de Salvatore J. Stolfo [6], *Behavior-based modeling and its application to Email analysis*, introduce Email Mining Toolkit (EMT), un sistema de minería de datos que construye modelos de comportamiento a partir del análisis de los correos electrónicos de los usuarios. Estos modelos pueden utilizarse para múltiples propósitos, incluyendo análisis forenses, detección de anomalías en la comunicación, seguridad informática y optimización organizativa. Un aspecto relevante de este estudio es su capacidad para detectar patrones de comportamiento en la comunicación digital, lo que permite identificar actividades inusuales, como la propagación de virus informáticos sin necesidad de análisis de contenido.

También existen casos de uso de empresas que ofrecen como servicio estos sistemas como es el caso de SuccessKP, una compañía que ha aplicado técnicas de Natural Language Processing (NLP) y machine learning para analizar comunicaciones internas dentro de una empresa. Su plataforma permite evaluar interacciones en tiempo real, detectar problemas en la comunicación y mejorar la experiencia del empleado. La empresa ha reportado mejoras en la eficiencia operativa de hasta un 10 % en compañías que han implementado sus soluciones, al reducir tiempos de respuesta y optimizar la colaboración entre departamentos.

Otras como Lattice, ofrecen un análisis organizacional permite a las empresas evaluar la comunicación interna y el rendimiento de los empleados mediante la recopilación y procesamiento de datos de correos electrónicos, herramientas de mensajería y plataformas de gestión de proyectos. Las empresas que han integrado Lattice en sus operaciones han logrado una reducción del 78 % en el tiempo dedicado a tareas administrativas.

Capítulo 3

Metodología

A lo largo de este capítulo, se describe la metodología que se seguirá para desarrollar el dashboard del proyecto, incluyendo el procesamiento de datos, análisis de sentimiento y visualización de resultados.

3.1. Aspectos técnicos

El código del proyecto será desarrollado en **Python** 3.11. Para la escritura y ejecución del código se utilizarán los entornos **Cursor** (versión 0.46.11 Universal) y **Google Colab**, aprovechando este último para disponer de una GPU en determinadas tareas. La gestión de librerías y la dockerización del proyecto se realizará con **Poetry**. Para la visualización del dashboard, se empleará **Dash**, un framework que permitirá desplegar aplicaciones web interactivas en un servidor local.

3.2. Procesamiento de datos

El procesamiento de datos se diseñará para limpiar, estructurar y analizar los correos electrónicos de la base de datos. A continuación, se detallan las principales etapas del proceso:

3.2.1. Carga y limpieza de datos

Los correos electrónicos se almacenarán en archivos CSV y se cargarán utilizando **pandas**. Se aplicarán las siguientes técnicas de limpieza:

- Eliminación de filas con encabezados incompletos.
- Extracción de información relevante como fecha, remitente, destinatarios y asunto.
- Conversión de fechas al formato estándar.
- Normalización y limpieza del cuerpo del mensaje eliminando caracteres no deseados.

3.2.2. Análisis de sentimiento

Para evaluar la polaridad de los correos electrónicos, se implementará un análisis de sentimiento utilizando dos enfoques:

- **VADER Sentiment Analyzer**: se empleará para clasificar los correos en positivos, negativos o neutrales en función del puntaje de sentimiento compuesto.
- **Modelo de Transformers**: se utilizará el modelo preentrenado `SamLowe` para detectar emociones en los mensajes.

3.2.3. Almacenamiento de datos procesados

Una vez procesados, los datos se almacenarán en un archivo CSV estructurado con las siguientes columnas clave:

- Fecha del correo.
- Remitente y destinatarios.
- Asunto.
- Cuerpo del mensaje limpio.
- Categoría de sentimiento.
- Emoción predominante.

3.3. Visualización en el dashboard

El dashboard interactivo será desarrollado utilizando **Dash** y **Plotly**, e incorporará diversas visualizaciones para analizar los correos electrónicos procesados.

3.3.1. Filtros y exploración de datos

El usuario podrá aplicar filtros para analizar los correos de interés según:

- Fecha de envío.
- Emoción predominante.
- Palabras clave dentro del mensaje.

3.3.2. Gráficos y análisis visual

Se implementarán varias visualizaciones interactivas:

- **Distribución de emociones**: Un gráfico de tipo pie mostrará la proporción de emociones detectadas en los correos electrónicos.
- **Red de comunicaciones**: Se construirá un grafo interactivo para visualizar las relaciones entre remitentes y destinatarios, con colores representando la polaridad del sentimiento.
- **Evolución temporal**: Un gráfico de líneas mostrará la evolución del volumen de correos a lo largo del tiempo.

3.4. Despliegue y ejecución

El dashboard se ejecutará localmente y permitirá la interacción en tiempo real con los datos procesados. La aplicación se iniciará mediante un servidor en local con Dash y estará optimizada para una exploración eficiente de la información extraída de los correos electrónicos.

Capítulo 4

Tratamiento de Datos

4.1. Descripción del *Enron Email Dataset*

Enron Email Dataset es una recopilación de aproximadamente 500,000 correos electrónicos generados por empleados de la corporación Enron antes de su quiebra en diciembre de 2001[9]. Este conjunto de datos ha sido ampliamente utilizado en investigaciones relacionadas con el procesamiento de lenguaje natural y el análisis de redes sociales.

Este dataset fue obtenido por la Comisión Federal Reguladora de Energía durante su investigación sobre el colapso de Enron.

4.1.1. Estructura del Conjunto de Datos

Los correos electrónicos están organizados en directorios correspondientes a 150 usuarios, principalmente de la alta gerencia de Enron. Cada directorio de usuario contiene subcarpetas que reflejan la estructura de carpetas de su cliente de correo electrónico, tales como `inbox`, `sent_items`, `deleted_items`, entre otras. A continuación, se muestra una tabla con las columnas del conjunto de datos:

Columna	**Descripción**
`file`	Ruta del archivo que contiene el correo electrónico dentro del directorio del usuario.
`message`	Contenido completo del correo electrónico, incluyendo encabezados y cuerpo del mensaje.

Cuadro 4.1: Descripción de columnas para correos electrónicos

4.1.2. Ejemplo de un Objeto `email`

A continuación, se presenta un ejemplo del contenido de un correo electrónico tal como aparece en el conjunto de datos:

```
Message-ID: <18782981.1075855378110.JavaMail.evans@thyme>
Date: Mon, 14 May 2001 16:39:00 -0700 (PDT)
From: phillip.allen@enron.com
To: tim.belden@enron.com
Subject: Futures Nomination

Tim love the worked you have done in the gas project. Adding you as a Future \
    Nomination.
```

Figura 4.1: Ejemplo correo del dataset de Enron

4.2. Procesamiento de los Correos Electrónicos

Para analizar los correos electrónicos, es necesario convertirlos desde su formato original a una estructura que facilite su manipulación y análisis.

4.2.1. Conversión del Objeto `email` a un Formato Legible

Los correos electrónicos en el conjunto de datos están en formato `RFC 822`[1], que incluye múltiples campos de encabezado y el cuerpo del mensaje. Para convertir estos correos a un formato más manejable, se seguirán los siguientes pasos:

1. **Extracción de Componentes:** Utilizando librerías de Python como `re` y `pandas`, se extraen los distintos campos relevantes del correo (tales como `Date`, `From`, `To`, `Subject` y el cuerpo del mensaje). Este paso se ve facilitado por el hecho de que todos los correos respetan una estructura común basada en el estándar RFC 822.

2. **Estructuración de los Datos:** Una vez extraídos, los componentes del correo se almacenan en un formato estructurado, como un `DataFrame` de `pandas`. Esto permite realizar análisis posteriores de forma eficiente, incluyendo tareas como la clasificación de sentimientos, la extracción de emociones o el análisis de redes entre usuarios.

Ejemplo Objeto Tratado

A continuación, se muestra cómo se transformaría el ejemplo anterior 4.1:

- **Fecha:** `2001-05-14 23:39:00+00:00`
- **Remitente:** `phillip.allen@enron.com`
- **Destinatario 1:** `tim.belden@enron.com`
- **Destinatario 2:**
- **Destinatario 3:**

[1]El formato RFC 822 es un estándar que define la estructura de los mensajes de correo electrónico. Un mensaje en este formato consta de un encabezado —con campos como `From`, `To`, `Subject` o `Date`— seguido por una línea en blanco y, a continuación, el cuerpo del mensaje. Es un formato de texto plano que representa cómo se transmiten los correos en la red.

- **Asunto:** `Futures Nomination`
- **Texto:** `Tim love the worked you have done in the gas project. Adding you as a Future Nomination.`

4.3. Limpieza de Datos

Antes de realizar cualquier análisis, es crucial limpiar el conjunto de datos para eliminar información irrelevante o errónea que pueda sesgar los resultados.

4.3.1. Eliminación de Datos No Relevantes

Como parte del proceso de depuración, se eliminaron aquellos correos electrónicos que no contenían encabezados esenciales como `Date`, `From`, `To`, `Subject` o `Message-ID`. Estos campos son fundamentales para poder estructurar, analizar y contextualizar cada mensaje de manera adecuada. La ausencia de alguno de ellos imposibilita un tratamiento fiable del contenido, por lo que dichos correos se consideraron no aptos para el análisis.

Tras aplicar este filtro, se eliminó un 21.54 % del total de correos (517.401 correos), conservando únicamente aquellos con la estructura completa requerida para el posterior procesamiento.

Capítulo 5

Procesamiento

Una vez que los datos han sido limpiados y estructurados correctamente, se procede a aplicar técnicas de *Natural Language Processing* (NLP) con el objetivo de extraer información semántica relevante de los correos electrónicos. Concretamente, se busca identificar tanto el **sentimiento general** como las **emociones específicas** presentes en los mensajes, lo que permitirá construir una visión más profunda del tono y el clima organizacional dentro de la empresa.

A continuación, se detallan los pasos seguidos en este análisis:

Análisis de Sentimiento con VADER

Para la detección del sentimiento general de los correos, se emplea la herramienta `VADER` (Valence Aware Dictionary and Sentiment Reasoner). VADER es un modelo léxico basado en reglas heurísticas, diseñado para analizar textos informales como los encontrados en redes sociales, foros o correos electrónicos. A diferencia de modelos complejos basados en aprendizaje profundo, `VADER` no utiliza una arquitectura *transformer* ni una red neuronal: en su lugar, emplea un diccionario de palabras con puntuaciones de valencia emocional y una serie de reglas gramaticales para ajustar dichas puntuaciones en función del contexto (por ejemplo, mayúsculas, signos de exclamación, adverbios intensificadores o negaciones).

El análisis se realiza mediante el siguiente fragmento de código:

```
analyzer = SentimentIntensityAnalyzer()
self.data['sentiment scores'] = self.data['text'].apply(
    lambda message: analyzer.polarity_scores(message)
)
self.data['Sentiment'] = self.data['sentiment scores'].apply(
    lambda x: 'Positive' if x['compound'] > 0.05 else (
        'Negative' if x['compound'] < -0.05 else 'Neutral'
    )
)
```

Listing 5.1: Clasificación de sentimiento con VADER

Primero, se instancia el analizador de sentimiento. Luego, para cada mensaje, se calcula una puntuación compuesta (`compound`) que refleja el tono emocional global. Esta puntuación se interpreta del siguiente modo:

- Si es mayor a 0.01, el mensaje se clasifica como **positivo**.
- Si es menor a -0.01, se clasifica como **negativo**.

- Si se encuentra entre ambos valores, se considera **neutral**.

Análisis de Emociones con Transformers

Más allá del sentimiento global, se aplica un análisis emocional más detallado utilizando modelos de aprendizaje profundo basados en *transformers*. En concreto, se ha empleado el modelo `roberta-base-go_emotions`, alojado en la plataforma Hugging Face y entrenado sobre el conjunto de datos `GoEmotions` de Google.

Este modelo ha sido entrenado para reconocer 22 categorías emocionales distintas, como *admiración*, *alegría*, *enfado*, *tristeza*, *gratitud*, *miedo*, entre otras. Gracias a su arquitectura basada en `RoBERTa` (una variante de BERT optimizada para mejor rendimiento), es capaz de capturar matices semánticos complejos en el lenguaje humano.

El modelo se carga y ejecuta mediante la función `pipeline`, como se muestra a continuación:

```
classifier = pipeline(
    task="text-classification",
    model="SamLowe/roberta-base-go_emotions",
    top_k=1,
    device=-1
)
```

Listing 5.2: Clasificación emocional con modelo RoBERTa

Donde:

- `top_k=1` indica que se selecciona únicamente la emoción con mayor probabilidad.
- `device=-1` especifica que el modelo se ejecutará en CPU (puede cambiarse a `device=0` para utilizar GPU si está disponible).

Ejemplo de salida tras el análisis:

```
Message-ID: <18782981.1075855378110.JavaMail.evans@thyme>
Date: Mon, 14 May 2001 16:39:00 -0700 (PDT)
From: phillip.allen@enron.com
To: tim.belden@enron.com
Subject: Futures Nomination

Tim love the work you have done in the gas project. Adding you as a Future \
    Nomination.
```

- **Sentimiento:** Positivo
- **Emoción:** Admiración

Procesamiento Final y Embeddings

Una vez realizados ambos análisis, los correos electrónicos cuentan con información enriquecida que incluye sentimiento, emoción predominante y texto limpio. Esta información resulta fundamental para su uso en aplicaciones posteriores como motores de búsqueda semánticos (*Retrieval-Augmented Generation*, RAG), construcción de grafos de comunicación o generación de dashboards.

Una vez ya hemos completado la limpieza y procesamiento de los datos, ahora vamos a diseñar el dashboard interactivo. Si se desea ver el resto del código, se ha adjuntado en el Anexo A.

Capítulo 6

Dashboard

Dashboard Interactivo para el Análisis Ejecutivo

Con el objetivo de transformar el análisis técnico de los correos electrónicos en una herramienta visual y accesible para la toma de decisiones, se ha desarrollado un **dashboard interactivo** basado en los datos previamente tratados. Esta plataforma permite a los responsables de la empresa explorar, interpretar y visualizar patrones de comunicación y emociones detectadas en los mensajes corporativos de manera intuitiva.

Tecnología utilizada: Dash

Para el desarrollo de esta herramienta se ha utilizado `Dash`, un framework de código abierto que permite construir aplicaciones web interactivas directamente en Python. A diferencia de otras tecnologías más complejas, Dash facilita la creación de interfaces profesionales sin necesidad de escribir código en HTML, CSS o JavaScript.

En este proyecto, Dash se ha combinado con bibliotecas especializadas en visualización y análisis de datos como `Plotly`, `pandas`, `NetworkX` y `Dash Bootstrap Components` para ofrecer un resultado final completo y profesional.

Estructura del Dashboard

El dashboard está dividido en tres secciones principales: **Overview**, **Network Analysis** y **Content Analysis**, cada una enfocada en una dimensión diferente del análisis de los correos.

1. Overview: Visión General

Esta pestaña ofrece una panorámica general de la actividad de correos electrónicos dentro de la organización:

- **Indicadores clave (KPIs):** Se muestran métricas resumen como el número total de comunicaciones, el sentimiento predominante, la emoción más frecuente y el comunicador más activo (Figura 7).
- **Filtros interactivos:** Permiten filtrar el contenido por texto, sentimiento, emoción o rango de fechas.
- **Evolución temporal:** Muestra la cantidad de correos enviados a lo largo del tiempo.

- **Distribución de emociones:** Un gráfico de barras muestra la frecuencia de cada emoción detectada.
- **Participantes clave:** Se identifican los empleados con mayor volumen de mensajes enviados o recibidos.
- **Visor de correos:** Permite seleccionar y visualizar correos individuales, junto con su sentimiento y emoción detectada.

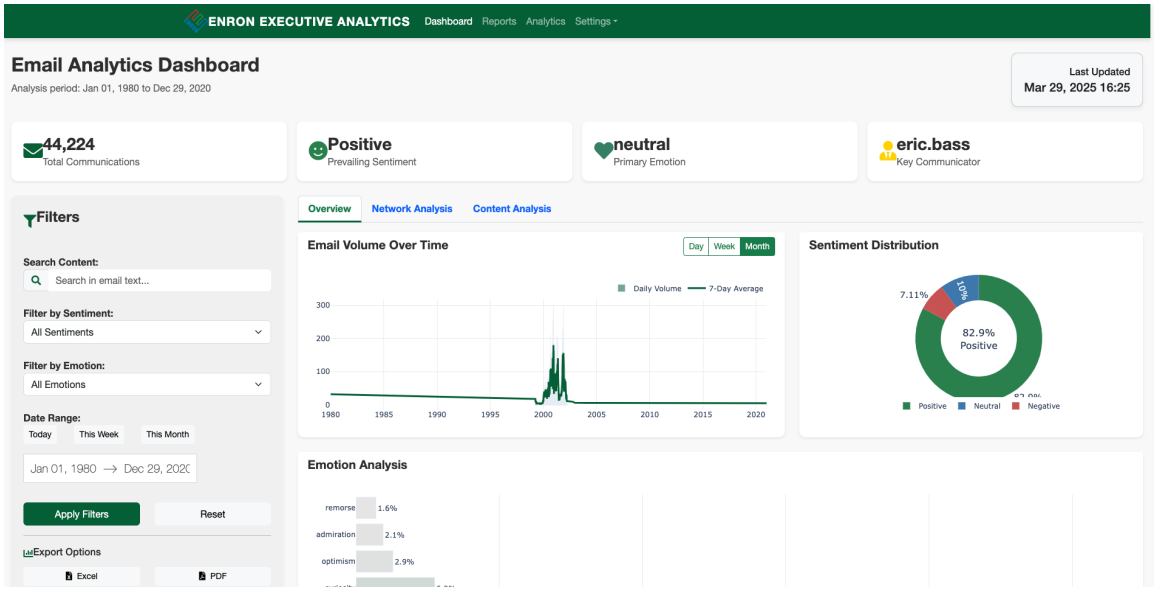

Figura 6.1: Resumen Visión General de las comunicaciones internas

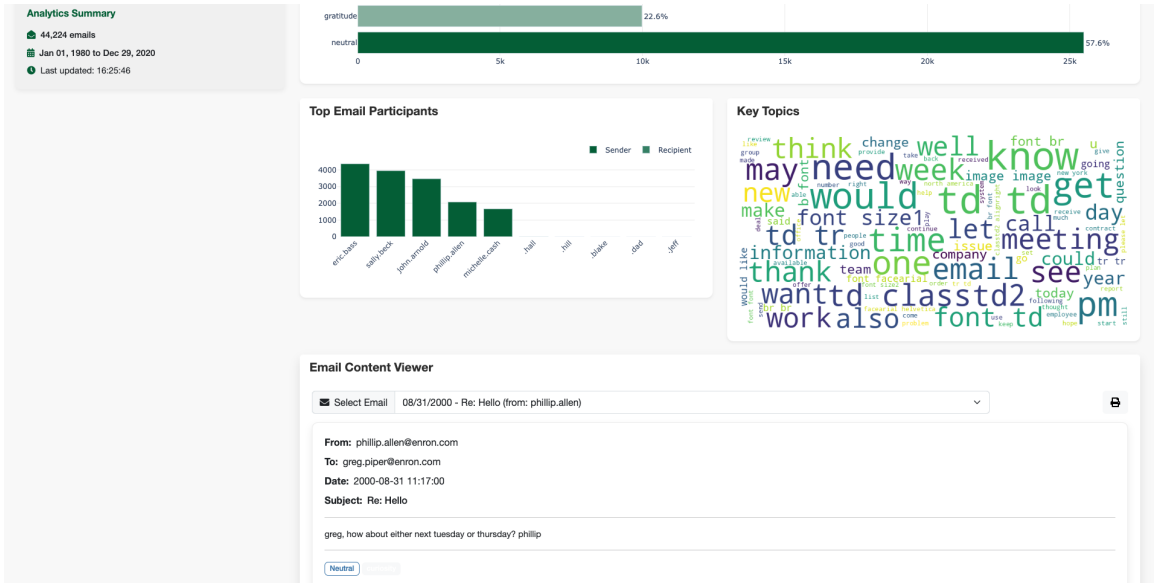

Figura 6.2: Resumen Visión General de las comunicaciones internas

2. Network Analysis: Red de Comunicación

Esta sección permite visualizar cómo se conectan entre sí los distintos empleados a través de sus correos:

- **Grafo de comunicación:** Cada nodo representa una persona y cada conexión un vínculo comunicativo. El grosor de las líneas indica la frecuencia de los correos.
- **Métricas de red:** Se calcula el número de conexiones clave, la densidad de la red y la longitud media de los caminos de comunicación.
- Esta herramienta ayuda a detectar empleados clave, cuellos de botella o posibles comunidades internas.

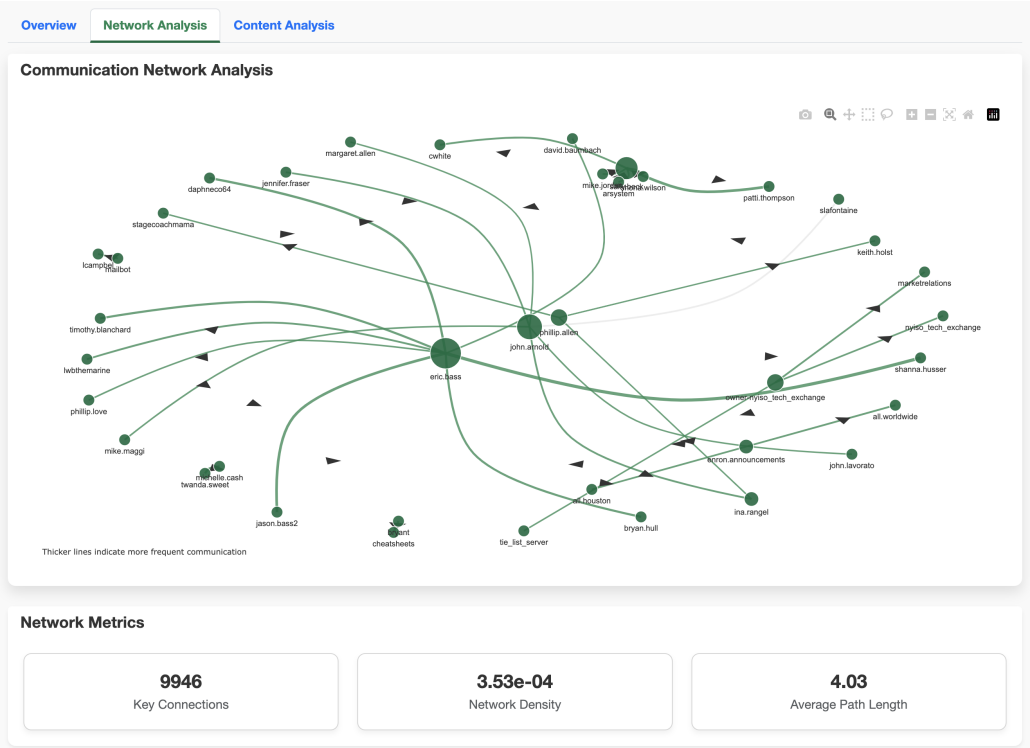

Figura 6.3: Visualización de la red de comunicaciones internas (Network Analysis)

Fórmulas y significado de las métricas de red:

- **Key Connections (Conexiones clave):** Número total de aristas (conexiones directas) en la red de comunicación.

$$\text{Key Connections} = |E|$$

  Donde $|E|$ representa el número total de conexiones directas (aristas) en el grafo.

- **Network Density (Densidad de la red):** Mide cuán conectada está la red en relación con el número máximo de conexiones posibles. Toma valores entre 0 y 1.

$$\text{Density} = \frac{2|E|}{|V|(|V|-1)}$$

  Donde $|V|$ es el número de nodos (personas) y $|E|$ es el número de conexiones (aristas).

- **Average Path Length (Longitud media del camino):** Promedio de las distancias más cortas entre todos los pares de nodos. Refleja la eficiencia del flujo de información en la red.

$$\text{Average Path Length} = \frac{1}{\frac{|V|(|V|-1)}{2}} \sum_{i \neq j} d(v_i, v_j)$$

Donde $d(v_i, v_j)$ es la distancia más corta entre los nodos v_i y v_j.

3. Content Analysis: Análisis del Contenido

La pestaña de análisis de contenido se enfoca en estudiar las palabras clave utilizadas y su carga emocional:

- **Nube de palabras:** Muestra los términos más utilizados en los correos, donde el tamaño indica su frecuencia.
- **Palabras más frecuentes:** Un gráfico de barras resume las palabras clave más repetidas.
- **Sentimiento por palabra:** Se analiza el sentimiento dominante asociado a cada palabra.
- **Evolución temporal del contenido:** Muestra cómo varían en el tiempo las menciones a ciertos términos clave.

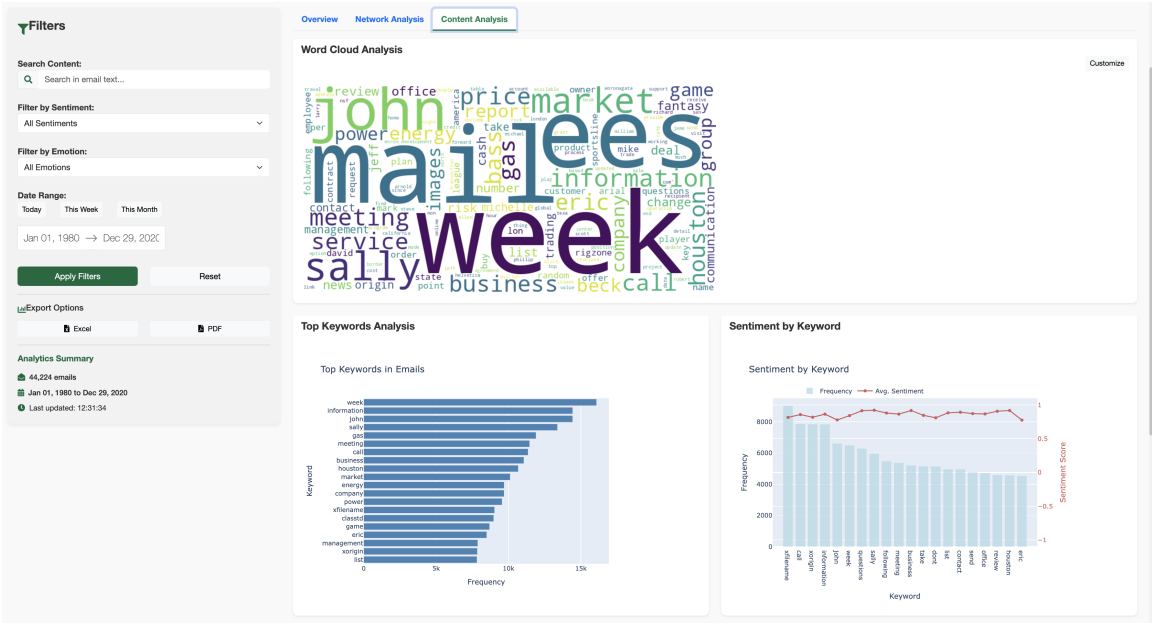

Figura 6.4: Análisis de contenido: nube de palabras, keywords y sentimiento asociado

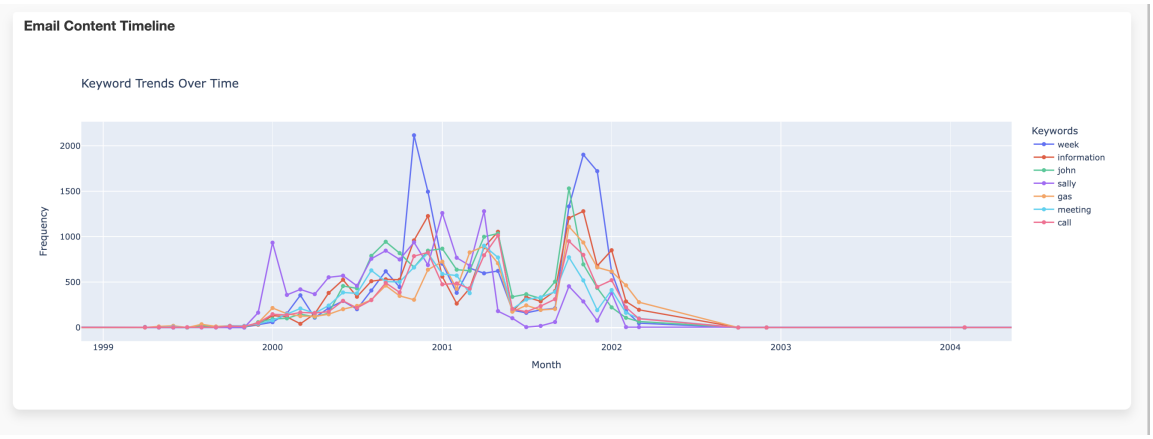

Figura 6.5: Tendencias de palabras clave a lo largo del tiempo

Utilidad para la Empresa

En el siguiente capítulo se hará un analisi de los resultados obtenidos, pero algunas de las capacidades más destacadas incluyen:

- Detectar cambios en el tono emocional de la comunicación durante periodos específicos, como crisis, reestructuraciones o lanzamientos.
- Identificar empleados o departamentos con una alta carga comunicativa, lo cual puede señalar roles clave o posibles cuellos de botella.
- Analizar las emociones predominantes asociadas a determinados contextos, ayudando a comprender el clima organizacional.
- Extraer palabras clave asociadas a estrés, conflictos, colaboración o agradecimiento, aportando una visión cualitativa del contenido de los correos.
- Utilizar un motor de búsqueda temático para localizar rápidamente a las personas más vinculadas a un tema concreto (por ejemplo, buscar *"trading de gas"* y conocer qué empleados discuten dicho tema para contactar directamente con ellos).

Capítulo 7

Análisis de Resultados

Este capítulo presenta un análisis detallado de las comunicaciones internas de Enron utilizando la herramienta desarrollada. El objetivo principal es detectar posibles ineficiencias estructurales, dinámicas organizativas y patrones de comportamiento. Lejos de limitarse a un ejercicio retrospectivo, este enfoque demuestra el valor de la herramienta como sistema de alerta temprana y como apoyo continuo en la toma de decisiones estratégicas.

Visión General del Panel

Los resultados iniciales revelan un sentimiento global mayoritariamente positivo (*82.9 %*) o neutro (*10.0 %*) en los correos electrónicos analizados. Este patrón puede interpretarse desde distintas perspectivas. Por un lado, podría reflejar una cultura corporativa basada en el optimismo, la cortesía y el lenguaje institucional positivo. Por otro, resulta llamativo en el contexto de una empresa que atravesaba una grave crisis, lo cual sugiere la existencia de mecanismos de contención emocional o incluso de censura implícita.

La ausencia de términos directamente asociados al escándalo financiero (*fraud*, *bankruptcy*, *SEC*) refuerza esta hipótesis. Es plausible pensar que los empleados, conscientes de que sus comunicaciones podían ser monitorizadas o utilizadas como prueba en procesos judiciales, optasen por evitar referencias explícitas a temas delicados. En su lugar, empleaban eufemismos, términos genéricos o desplazaban las conversaciones más sensibles a canales menos rastreables, como reuniones presenciales o llamadas telefónicas. Esta dinámica constituye en sí misma un indicador de la cultura organizativa de la compañía y puede ser interpretada como una forma de autocensura motivada por el miedo a posibles repercusiones legales.

Análisis de Contenido: Nube de Palabras

La generación de la nube de palabras, tras un proceso riguroso de limpieza de *stopwords*, permite identificar los ejes temáticos dominantes:

- **Finanzas y comercio**: términos como *market*, *deal*, *contract*, *trade*, reflejan una actividad comercial intensa, centrada en la compraventa de productos energéticos.
- **Sector energético**: palabras como *gas*, *energy*, *houston*, *rigzone* reafirman el núcleo del negocio de Enron.

- **Gestión interna**: la aparición frecuente de *meeting*, *plan*, *report* sugiere una estructura organizativa orientada a la planificación constante.
- **Dimensión internacional**: la mención de localizaciones como *california*, *london* o *america* confirma el alcance global de la compañía.
- **Verbos de acción**: como *send*, *request*, *support*, refuerzan el uso operativo del canal de correo electrónico.
- **Nombres propios**: como *john*, *sally*, *jeff*, fundamentales para la construcción de redes de comunicación internas.
- **Otros**: términos como *fantasy* o *game* podrían asociarse a correos informales o no corporativos.

Identificación de Actores Clave

Una de las funcionalidades más relevantes de la herramienta es la detección de nodos con alta centralidad en la red comunicativa. Destaca el caso de **Eric Bass** (el cual, con un análisis en internet, se ha visto que era un coordinador entre diferentes áreas.), quien presenta la mayor carga comunicativa del corpus. Su posición sugiere un rol informal de liderazgo o coordinación transversal. Esta información resulta valiosa para la gestión de talento y la prevención de cuellos de botella operativos.

Otros actores relevantes como Sally Beck (COO), John Arnold (Un trader con alta reputación dentro de la empresa) y Phillip Allen (dentro de la división de trading) también muestran alta centralidad. La monitorización de estas figuras puede ser crucial tanto para la planificación de sucesiones como para el análisis de riesgo organizativo, especialmente en estructuras donde el conocimiento está altamente concentrado.

Análisis de la Red de Comunicación

El grafo de comunicaciones revela una estructura organizativa caracterizada por una **baja densidad** (**0.00035**) y una **longitud media de los caminos de 4.03**. En términos prácticos, esto implica que un mensaje debe recorrer, en promedio, más de cuatro intermediarios para llegar a su destinatario, lo que evidencia una arquitectura jerárquica con escasa transversalidad entre equipos.

Este patrón estructural se asocia comúnmente con organizaciones de gran tamaño y conlleva implicaciones relevantes:

- **Alta dependencia de nodos centrales**, cuya desconexión puede afectar de forma significativa la eficiencia comunicativa.
- **Posibles cuellos de botella**, dado que ciertos individuos o departamentos concentran un volumen desproporcionado de tráfico informativo.
- **Limitada fluidez en la transmisión de información interdepartamental**, lo que puede dificultar la coordinación y la toma de decisiones ágiles.

Estos elementos sugieren la necesidad de estrategias que fomenten la descentralización de la comunicación y la creación de canales transversales más eficientes. Como propuesta de mejora, la organización podría:

- Diseñar eventos corporativos estratégicos para reforzar conexiones transversales.
- Reducir la dependencia de individuos clave mediante redistribución de responsabilidades.
- Optimizar los flujos de comunicación para aumentar la resiliencia organizativa.

Análisis Temporal y Semántico

El análisis temporal de términos clave y métricas de sentimiento revela dinámicas interesantes:

- A partir de 2001 se incrementa el uso de palabras asociadas a la planificación (*meeting*, *call*), lo que podría reflejar intentos de reorganización ante una creciente presión interna.
- El sentimiento se mantiene sorprendentemente positivo, lo cual refuerza la hipótesis de autocensura o desplazamiento de las conversaciones a canales no digitales.
- Tras 2002, se observa una caída abrupta en el volumen de correos, en línea con la disolución de equipos y el colapso operativo.

Además, se identifican picos de actividad comunicativa antes de fechas clave como **earnings reports**, lo que sugiere una mayor carga de trabajo y coordinación en momentos críticos. Esta información permite evaluar el estrés organizativo y anticipar necesidades de refuerzo en ciertos departamentos o perfiles.

Evaluación Estratégica

Más allá de su utilidad como herramienta forense, esta plataforma tiene un enorme potencial como sistema de medición del impacto de cambios organizativos. Por ejemplo:

- Tras la implementación de una política de comunicación transversal, podría observarse una reducción en la longitud media de los caminos o un aumento en la densidad de la red.
- Ante la rotación de un perfil clave, el análisis de centralidad puede servir para evaluar el impacto en tiempo real.
- La identificación de zonas con baja interacción puede guiar estrategias de integración, formación o cambio cultural.
- A través del buscador, hacer un filtrado por un tema en específico y ver como cambia la estructura.

Reflexión final del ejemplo

El caso Enron muestra cómo la ausencia de señales explícitas puede ser indicativa de un entorno de control y opacidad. La herramienta desarrollada habría permitido detectar, entre otros:

- Concentración excesiva de poder en ciertos nodos comunicativos.
- Cambios abruptos en los patrones de planificación o coordinación.
- Ausencia de referencias a temas críticos, lo que puede interpretarse como censura implícita.
- Falta de robustez en la red de comunicación.

En definitiva, este tipo de análisis demuestra cómo el estudio de metadatos y contenido textual puede aportar un **valor estratégico real** en la identificación de ineficiencias, prevención de crisis y mejora continua de las dinámicas internas de una organización.

Capítulo 8

Conclusión

A lo largo de este trabajo se ha desarrollado un sistema completo de análisis basado en técnicas de Natural Language Processing (NLP), aplicado al estudio de las comunicaciones internas de la empresa Enron. Desde la extracción y depuración de datos hasta su análisis e interpretación visual, se ha construido un dashboard interactivo que permite explorar, comprender y extraer valor de grandes volúmenes de correos electrónicos. Este proyecto demuestra cómo la combinación de técnicas avanzadas de procesamiento de lenguaje con herramientas de visualización puede convertirse en una solución eficaz para apoyar la toma de decisiones estratégicas en entornos corporativos complejos.

Entre los principales logros alcanzados, destacan los siguientes:

- **Depuración y Estructuración de Datos:** Se implementó un proceso robusto de limpieza y normalización de correos electrónicos, que garantiza la coherencia del dataset y permite un análisis fiable.
- **Análisis de Sentimiento y Emociones:** Se utilizó una doble aproximación: la herramienta VADER para el análisis de sentimiento y un modelo basado en `Transformers` (`RoBERTa` entrenado con `GoEmotions`) para la identificación precisa de emociones. Esta combinación aporta una visión rica y matizada del clima emocional de la organización.
- **Visualización Interactiva de Resultados:** A través de tecnologías como `Dash`, `Plotly` y `NetworkX`, se desarrolló una interfaz accesible que permite:
  - Identificar los flujos de comunicación más relevantes dentro de la organización.
  - Analizar la evolución emocional a lo largo del tiempo.
  - Detectar nodos clave, cuellos de botella y patrones de interacción inusuales.
  - Explorar el contenido mediante filtros personalizados por fechas, emociones, palabras clave o remitentes.

Futuras Líneas de Mejora

Aunque el sistema desarrollado ofrece resultados satisfactorios, existen varias vías para su ampliación y refinamiento:

- **Entrenamiento de un Modelo Propio de Emociones:** Entrenar un *transformer* específico para el dominio de la empresa (p. ej., *emails corporativos* con categorías

emocionales más personalizadas) podría mejorar la precisión de la detección de emociones.

- **Integración de Modelos de Lenguaje a Gran Escala (LLM):** Herramientas como `GPT` o `BERT` en su versión *fine-tuned* podrían utilizarse para generar resúmenes automáticos de los hilos de conversación o extraer acciones clave (*action items*) de cada correo.
- **Implementación de un Sistema RAG (Retrieval-Augmented Generation):** Al combinar un LLM con una base de datos de vectores, como `Pinecone`, sería posible realizar búsquedas semánticas avanzadas, localizando automáticamente la información más relevante y permitiendo la generación de respuestas contextuales de alta precisión.
- **Análisis de Sentimiento Multilingüe o Más Fino:** En caso de tener correos en múltiples idiomas, o si se desea una categorización más detallada del sentimiento, podrían emplearse modelos y diccionarios específicos para cada lengua, o bien integrar sistemas de clasificación de sentimiento por *aspectos* (aspect-based sentiment analysis).

Conclusión General

El presente trabajo ha puesto de manifiesto el enorme potencial del análisis automatizado del lenguaje en el ámbito empresarial. La posibilidad de analizar, de forma sistemática y comprensible, la red de comunicaciones internas y los estados emocionales que la atraviesan, constituye una herramienta estratégica de alto valor. El desarrollo del dashboard no solo permite detectar patrones relevantes, sino también democratizar el acceso a esta información, empoderando a perfiles no técnicos mediante visualizaciones intuitivas y filtros accesibles.

En definitiva, esta solución constituye una base sólida sobre la que construir herramientas aún más potentes. Con su capacidad para escalar, adaptarse a distintos contextos organizativos y enriquecer la toma de decisiones con evidencia objetiva, representa un paso firme hacia una gestión empresarial más inteligente, empática y basada en datos.

Apéndice A

Limpieza_data.py

```
import pandas as pd
import numpy as np
import re
import os
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from transformers import pipeline
from tqdm import tqdm
import torch


class EmailProcessor:
    def __init__(self, file_path, max_rows=None, header_lines=15):
        self.file_path = file_path
        self.max_rows = max_rows
        self.header_lines = header_lines
        self.data = None

    def load_data(self):
        """Carga el archivo CSV y almacena los datos en un DataFrame."""
        try:
            if self.max_rows is None:
                self.data = pd.read_csv(self.file_path)
            else:
                self.data = pd.read_csv(self.file_path, nrows=self.max_rows)
            print(f"Datos cargados correctamente desde {self.file_path}.")
        except Exception as e:
            print(f"Error al cargar los datos: {e}")

    @staticmethod
    def extract_text(series, row_num_slicer):
        return series.apply(lambda msg:
"\n".join(msg.split("\n")[row_num_slicer:]).strip())

    @staticmethod
    def extract_row(series, row_num):
        return series.apply(lambda msg:
msg.split("\n")[row_num].strip() if len(msg.split("\n")) > row_num
else "")

    @staticmethod
    def extract_addresses(series, num_cols=1):
        email_regex = r"([\w\.-]+@[\w\.-]+\.\w+)"
```

```
        all_matches = series.str.extractall(email_regex)

        extracted = all_matches.unstack(fill_value="")

        # Renombrar columnas numeradas si hay ms de num_cols
        extracted.columns = extracted.columns.droplevel(0)

        # Ajustar el nmero de columnas
        actual_cols = list(extracted.columns)
        if len(actual_cols) < num_cols:
            # Rellenar columnas que faltan con strings vacos
            for i in range(len(actual_cols), num_cols):
                extracted[i] = ""
        elif len(actual_cols) > num_cols:
            # Cortar columnas sobrantes
            extracted = extracted[[i for i in range(num_cols)]]

        extracted.columns = [f"email_{i+1}" for i in range(num_cols)]

        return [extracted[col] for col in extracted.columns]


    @staticmethod
    def clean_body_text(series):
        pattern = re.compile(r"[<>\n\t\s\*]+")
        cleaned = series.str.replace(pattern, " ", regex=True)
        cleaned = cleaned.str.replace("\r", "", regex=False)
        return cleaned.str.strip().str.lower()

    def clean_headers(self):
        headers = ["Message-ID: ", "Date: ", "From: ", "To: ", "Subject: "]
        initial_count = len(self.data)
        for i, header in enumerate(headers):
            self.data =
self.data[self.data["message"].str.split("\n").str[i].str.contains(header,
na=False, regex=False)]
        self.data.reset_index(drop=True, inplace=True)
        removed_count = initial_count - len(self.data)
        print(f"Eliminados {removed_count} correos sin encabezados
clave ({(removed_count / initial_count) * 100:.2f}%).")

    def process_emails(self, batch_size=64, emotion_limit=None,
checkpoint_path="emails_temp_emotions.csv"):
        if self.data is None:
            print("No hay datos cargados. Llama a load_data() primero.")
            return

        print("Iniciando procesamiento de correos...")

        self.data["text"] = self.extract_text(self.data["message"],
self.header_lines)
        self.data["date"] = self.extract_row(self.data["message"],
1).str.replace("Date: ", "", regex=False)
        self.data["senders"] = self.extract_row(self.data["message"], 2)
        self.data["recipients"] = self.extract_row(self.data["message"], 3)
        self.data["subject"] = self.extract_row(self.data["message"],
4).str.replace("Subject: ", "", regex=False)
        self.data["date"] = pd.to_datetime(self.data["date"],
```

```
errors="coerce", utc=True)


        self.data["recipient1"], self.data["recipient2"],
self.data["recipient3"] = \
            self.extract_addresses(self.data["recipients"], 3)
        self.data["sender"], _, _ =
self.extract_addresses(self.data["senders"], 3)


        self.data["text"] = self.clean_body_text(self.data["text"])

        tqdm.pandas(desc="Analizando sentimiento (VADER)")
        analyzer = SentimentIntensityAnalyzer()
        self.data['sentiment scores'] =
self.data['text'].progress_apply(lambda message:
analyzer.polarity_scores(message))
        self.data['Sentiment'] = self.data['sentiment scores'].apply(
            lambda x: 'Positive' if x['compound'] > 0.05 else
('Negative' if x['compound'] < -0.05 else 'Neutral')
        )

        # === EMOCIONES CON TRANSFORMERS (optimizado) ===
        if os.path.exists(checkpoint_path):
            print("Cargando checkpoint de emociones existente...")
            self.data = pd.read_csv(checkpoint_path)
            return

        print("Cargando modelo de anlisis de emociones...")
        device = 0 if torch.cuda.is_available() else -1
        classifier = pipeline(
            task="text-classification",
            model="SamLowe/roberta-base-go_emotions",
            top_k=1,
            device=device
        )
        print(f"Usando {'GPU' if device == 0 else 'CPU'} para anlisis
de emociones.")

        # Limitacin opcional para debug
        if emotion_limit:
            self.data = self.data.head(emotion_limit)

        all_texts = self.data["text"].tolist()
        emotions = []

        def get_top_label(result_list):
            result_list.sort(key=lambda x: x['score'], reverse=True)
            return result_list[0]['label'] if result_list else "N/A"

        print("Analizando emociones (Transformers)...")
        for i in tqdm(range(0, len(all_texts), batch_size),
desc="Analizando emociones"):
            batch = all_texts[i:i+batch_size]
            batch = [text[:800] for text in batch]  # truncar si es necesario
            results = classifier(batch)
            emotions.extend([get_top_label(r) for r in results])

```

```
            # Guardar progreso intermedio cada 10 batches
            if i % (batch_size * 10) == 0:
                temp_df = self.data.iloc[:i+batch_size].copy()
                temp_df["Emotions"] = emotions
                temp_df.to_csv(checkpoint_path, index=False)

        self.data['Emotions'] = emotions

        # Guardar checkpoint final
        self.data.to_csv(checkpoint_path, index=False)

        self.data.drop(columns=["recipients", "senders", "message",
"sentiment scores"], inplace=True)
        self.data = self.data[[
            "date", "sender", "recipient1", "recipient2", "recipient3",
            "subject", "text", "Sentiment", "Emotions"
        ]]
        print("Procesamiento finalizado.")

    def save_to_csv(self, output_path):
        try:
            self.data.to_csv(output_path, index=False)
            print(f"Datos guardados en {output_path}.")
        except Exception as e:
            print(f"Error al guardar los datos: {e}")


# ------------------------------
# USO DEL PROCESADOR
# ------------------------------
file_path = "emails.csv"
processor = EmailProcessor(file_path)
processor.load_data()
processor.clean_headers()
processor.process_emails(batch_size=64, emotion_limit=None)  # Puedes
probar con emotion_limit=1000
processor.save_to_csv("emails_procesados.csv")
```

Listing A.1: Limpieza_data.py

Apéndice B

App.py

```python
import dash
from dash import dcc, html
import dash_bootstrap_components as dbc
from dash.exceptions import PreventUpdate
import os

# Import components
from components.navbar import create_navbar
from components.sidebar import create_sidebar
from layouts.main_layout import create_main_layout
from callbacks.filter_callbacks import register_filter_callbacks
from callbacks.overview_callbacks import register_overview_callbacks
from callbacks.network_callbacks import register_network_callbacks
from callbacks.content_callbacks import register_content_callbacks
from callbacks.tab_callbacks import register_tab_callbacks
from utils.data_utils import load_data

# Create assets folder if it doesn't exist
os.makedirs('assets', exist_ok=True)

# Initialize the app
app = dash.Dash(
    __name__,
    external_stylesheets=[dbc.themes.BOOTSTRAP,
                          'https://use.fontawesome.com/releases/v5.15.4/css/all.css'],
    meta_tags=[{"name": "viewport", "content": "width=device-width, \
    initial-scale=1"}],
    suppress_callback_exceptions=True
)
app.title = "Enron Executive Email Analytics"
server = app.server

# Load the data
df_global = load_data()

# Create the navbar
navbar = create_navbar()

# Set up the app layout
app.layout = create_main_layout(navbar, df_global)

# Register callbacks
```

```
register_filter_callbacks(app, df_global)
register_overview_callbacks(app)
register_network_callbacks(app)
register_content_callbacks(app)
register_tab_callbacks(app)  # Add this line for tab callbacks

# Run the app
if __name__ == '__main__':
    app.run(debug=False, threaded=True)
```

Listing B.1: App.py

Apéndice C

overview.py

```
# Callbacks for the content analysis tab

from dash import Input, Output, State
import pandas as pd
import plotly.graph_objects as go
from utils.text_utils import create_wordcloud
from utils.chart_utils import (
    create_keyword_frequency_chart,
    create_sentiment_by_keyword_chart,
    create_content_timeline_chart
)

ENRON_STOP_WORDS = [
    # Saludos y cortesa
    "thanks", "thank", "please", "dear", "regards", "sincerely", "hello",
    "hi", "hey", "best",

    # Encabezados y metadatos de email
    "email", "message", "forwarded", "original", "subject", "sent", "attachment",
    "attach", "attached", "file", "files", "document", "documents", "cc", "bcc", \
    "fw", "re",

    #  Das  de la semana
    "monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday",

    # Meses y abreviaturas
    "jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", \
    "dec",
    "january", "february", "march", "april", "june", "july", "august", "september",
    "october", "november", "december",

    # Verbos auxiliares y modales
    "would", "could", "should", "shall", "might", "may", "can", "will", "must",

    # Horas y tiempos
    "a.m", "p.m", "am", "pm", "date", "time", "today", "last", "day", "year", "month",

    # HTML, CSS, y artefactos web
    "td", "td2", "tr", "font", "size", "align", "class", "image", "img", "src", \
    "href",
    "html", "http", "https", "www", "net", "mailto", "filename", "click", "style",
    "color", "face", "width", "height", "asp", "gif",
```

```
    # Dominios y trminos genricos de red
    "com", "corp", "hou", "inc", "ltd",

    # Trminos muy genricos / bajo contenido semntico
    "get", "see", "also", "know", "right", "let", "like", "make", "back", "new",
    "one", "team", "system", "good", "well", "work", "issue", "want", "need", "use",
    "way", "two",

    # Palabras genricas aadidas ahora
    "said", "come", "still", "give", "start", "going", "help", "free", "next",
    "first", "look", "sure", "think", "question",

    # Restos de HTML o cdigo mal procesado
    "nbsp", "script", "note", "spacer", "site", "cgi",

    # Palabras personalizadas
    "sabes", "enron", "schedule", "etc", "ect"
]

def register_content_callbacks(app):
    """Register callbacks for the Content Analysis tab"""

    @app.callback(
        [Output("content-wordcloud", "src"),
         Output("keyword-frequency-chart", "figure"),
         Output("keyword-sentiment-chart", "figure"),
         Output("content-timeline-chart", "figure")],
        [Input("filtered-data-store", "data")],
        [State("main-tabs", "active_tab")]
    )
    def update_content_analysis(filtered_data, active_tab):
        if not filtered_data or active_tab != "tab-content":
            empty_fig = go.Figure()
            empty_fig.update_layout(title="No Data Available")
            return "", empty_fig, empty_fig, empty_fig

        filtered_df = pd.DataFrame(filtered_data)

        # Generate an enhanced wordcloud for the content tab with stop words filtering
        all_text = ' '.join(filtered_df['text'].fillna('').tolist())
        wordcloud_src = create_wordcloud(all_text, stop_words=ENRON_STOP_WORDS)

        # Create keyword frequency chart with stop words filtering
        keyword_fig = create_keyword_frequency_chart(filtered_df, \
    stop_words=ENRON_STOP_WORDS)

        # Create sentiment by keyword chart with stop words filtering
        sentiment_keyword_fig = create_sentiment_by_keyword_chart(filtered_df, \
    stop_words=ENRON_STOP_WORDS)

        # Create content timeline chart with stop words filtering
        timeline_fig = create_content_timeline_chart(filtered_df, \
    stop_words=ENRON_STOP_WORDS)

        return wordcloud_src, keyword_fig, sentiment_keyword_fig, timeline_fig
```

Listing C.1: overview.py

Declaración de Uso de Herramientas de Inteligencia Artificial Generativa en Trabajos Fin de Grado

ADVERTENCIA: Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, **NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código** porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, **Juan Carlos Vecino de Haro**, estudiante de **Ingeniería de Telecomunicaciones y Business Analytics** de la Universidad Pontificia Comillas, al presentar mi Trabajo Fin de Grado titulado **"Dashboard interactivo para el análisis organizacional en la comunicación empresarial"**, declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación *[he mantenido sólo aquellas en las que se ha usado ChatGPT o similares y borrado el resto]*:

1. Metodólogo: Para descubrir métodos aplicables a problemas específicos de investigación.
2. Interpretador de código: Para realizar análisis de datos preliminares.
3. Corrector de estilo literario y de lenguaje: Para mejorar la calidad linguística y estilística del texto.
4. Generador previo de diagramas de flujo y contenido: Para esbozar diagramas iniciales.
5. Revisor: Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
6. Traductor: Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG

y he explicitado para qué se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 10 de Abril 2025

Firma: Juan Vecino

Bibliografía

[1] M. Intelligence, «Análisis de participación y tamaño del mercado de tecnología de Big Data: tendencias y pronósticos de crecimiento (2024-2029),» Mordor Intelligence, inf. téc., 2024. dirección: https://www.mordorintelligence.com/es/industry-reports/fdi-perspective-of-big-data-technology.

[2] E. Covas, *Named entity recognition using GPT for identifying comparable companies*, 2023. arXiv: 2307.07420 [cs.CL]. dirección: https://arxiv.org/abs/2307.07420.

[3] A. G. Al-Ali, R. Phaal y D. Sull, *Deep Learning Framework for Measuring the Digital Strategy of Companies from Earnings Calls*, 2020. arXiv: 2010.12418 [cs.CL]. dirección: https://arxiv.org/abs/2010.12418.

[4] Q. Wen, P. A. Gloor, A. F. Colladon, P. Tickoo y T. Joshi, «Finding top performers through email patterns analysis,» *Journal of Information Science*, vol. 46, n.º 4, págs. 508-527, 2020. DOI: 10.1177/0165551519849519. eprint: https://doi.org/10.1177/0165551519849519. dirección: https://doi.org/10.1177/0165551519849519.

[5] F. B. Viégas, S. Golder y J. Donath, «Visualizing email content: portraying relationships from conversational histories,» en *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ép. CHI '06, Montréal, Québec, Canada: Association for Computing Machinery, 2006, ISBN: 1595933727. DOI: 10.1145/1124772.1124919. dirección: https://doi.org/10.1145/1124772.1124919.

[6] S. J. Stolfo, S. Hershkop, C.-W. Hu, W.-J. Li, O. Nimeskern y K. Wang, «Behavior-based modeling and its application to Email analysis,» *ACM Trans. Internet Technol.*, vol. 6, n.º 2, 2006, ISSN: 1533-5399. DOI: 10.1145/1149121.1149125. dirección: https://doi.org/10.1145/1149121.1149125.

[7] SuccessKPI, *SuccessKPI - AI-Powered Analytics for Customer Experience and Communication Insights*, 2025. dirección: https://successkpi.com/.

[8] Lattice, *Lattice - People Success Platform*, 2025. dirección: https://lattice.com/.

[9] William W. Cohen, *Enron Email Dataset*, 2015. dirección: https://www.cs.cmu.edu/~enron/.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado y J. Dean, *Distributed Representations of Words and Phrases and their Compositionality*, 2013. arXiv: 1310.4546 [cs.CL]. dirección: https://arxiv.org/abs/1310.4546.

[11] D. Foster, *Generative Deep Learning*, 2nd. O'Reilly Media, Inc., 2023, ISBN: 9781098134181. dirección: https://learning.oreilly.com/library/view/generative-deep-learning/9781098134181/.

[12] A. Vaswani, N. Shazeer, N. Parmar et al., *Attention Is All You Need*, 2023. arXiv: 1706.03762 [cs.CL]. dirección: https://arxiv.org/abs/1706.03762.

[13] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,» *arXiv preprint arXiv:1810.04805*, 2018. dirección: https://arxiv.org/abs/1810.04805.

[14] S. Lowe, *roberta-base-go$_e$motions − onnx*, https://huggingface.co/samlowe/roberta-base-go_emotions-onnx, Disponible en Hugging Face, 2023. dirección: https://huggingface.co/samlowe/roberta-base-go_emotions-onnx.