UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CUANTIFICAR LA COMPLEJIDAD DE LAS OPINIONES Y DEBATES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

IGNACIO ADOLFO DÍAZ LARA

PROFESOR GUÍA:
ANDRÉS ABELIUK KIMELMAN

MIEMBROS DE LA COMISIÓN:
JOSÉ PIQUER GARDNER
ALEJANDRO HEVIA ANGULO

SANTIAGO DE CHILE
2022

Resumen

En el presente trabajo de memoria se presenta el desarrollo e implementación de un método para cuantificar la complejidad de un texto en lenguaje natural especialmente enfocado a entender de mejor manera cómo representar las opiniones y debates. Cuantificar la complejidad es obtener la cantidad mínima de dimensiones en que se puede representar el texto. La motivación para este trabajo nace de lo insatisfactorio de las soluciones que simplifican los debates y opiniones a representaciones unidimensionales, potencialmente perdiendo mucha información.

El método desarrollado consiste en modelar las dimensiones como diferenciales semánticos, que es el espectro que hay entre dos representaciones vectoriales de palabras que simbolizan dos extremos en términos de significado. Utilizando el *framework POLAR* y los diferenciales semánticos, se representan las palabras o documentos en forma de *embeddings* polares, para finalmente reducir sus dimensiones con el método de análisis de componentes principales para un nivel de varianza representada dado, que sirve como medida de sensibilidad. El cambio de eje de coordenadas que implica el análisis de componentes principales entrega pesos por cada dimensión original, que ya que están basadas en dimensiones de los *embeddings* polares, pueden entregar interpretabilidad sobre las dimensiones de las opiniones o debates.

Al estudiar la solución en un *dataset* de noticias de la *BBC* se observó que es importante comparar textos en tamaños similares y que el método funciona como un comparador de complejidades entre documentos o conjuntos de documentos. Se mostró que niveles altos de varianza sirven para comparar entre documentos, y niveles más bajos de varianza pueden ser utilizados para comparar entre conjuntos de documentos. Además, se observó que la complejidad obtenida con *word embeddings* o *embeddings* polares está correlacionado y no se obtuvo mayor capacidad de interpretación de los pesos asociados a las dimensiones de los *embeddings* polares.

Agradecimientos

Quiero agradecer a mi Mamá, mi tía Mary, mi Abuela, mi Abuelo y toda mi familia. Agradecer especialmente a Belén. Agradecer a mis amigos y amigas. Agradecer al profe Andrés. Agradecer a todas las personas que trabajan en la facultad.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Introducción

Las opiniones son una parte fundamental de la sociedad humana, se encuentran en instancias formales como debates en los parlamentos de los países o en interacciones más cotidianas como las discusiones en foros de internet en las que puede participar cualquier persona. Estas discusiones pueden tener diferentes complejidades: a veces son dos bandos los que discuten y otras veces son múltiples las posiciones en que se sitúan las opiniones de las personas participantes.

Hoy en día se encuentran disponibles datos de discusiones y opiniones de todo tipo que se pueden procesar para extraer información. Por ejemplo, datos de declaraciones de políticos estadounidenses son utilizados para clasificar si corresponden a opiniones liberales o conservadoras [2]. Pero para lograr esto se asume que existen solamente esas dos categorías, es decir un enfoque unidimensional. Este enfoque de las discusiones no puede representar de manera completa a las opiniones de mayor complejidad que poseen componentes de categorías diferentes y que pueden ser tangenciales a las categorías propuestas. En el ejemplo anterior, podría considerarse una segunda dimensión como qué tan ecológica es la declaración de la persona y, así, obtener más información de la discusión o de una opinión en particular. Tomando esto en cuenta, conocer la complejidad de una discusión ayudaría a representarlas de mejor manera en un espacio dimensional.

Existe un recurso actual que permite estimar qué tan compleja es una discusión aproximando la complejidad de las opiniones como la cantidad de dimensiones latentes que las caracterizan pero que necesita tener etiquetadas las opiniones con votaciones de otras personas participantes en la discusión [18]. Esto lo hace utilizando datos de foros de internet en los que se puede agregar comentarios y valorar los comentarios de otras personas con me gusta o no me gusta.

En este contexto es que se propone desarrollar una herramienta que permita cuantificar la complejidad de las opiniones, debates u otro tipo de texto, sin la necesidad de contar con datos etiquetados de discusiones, de tal manera que la herramienta reciba texto en lenguaje natural y cuantifique su complejidad. Para esto se utilizarán herramientas de procesamiento de lenguaje natural.

Si bien el punto de partida es caracterizar opiniones o debates, el poder recibir cualquier texto en lenguaje natural permite la posibilidad de analizar la complejidad de noticias, columnas de opinión, libros, discursos, etc. Por lo que este trabajo de memoria podría tener utilidad para encontrar una mejor forma de caracterizar las opiniones que se dan en ambientes políticos (parlamentos, congresos, convención constituyente, etc.), en artículos de prensa,

en discusiones en redes sociales, entre otras. Si primero se entiende la complejidad de las opiniones que se están dando en una discusión, ésta se puede analizar teniendo en cuenta el mínimo de dimensiones en que se puede representar una opinión, luego realizar un análisis que no reduzca la dimensionalidad y, por lo tanto, que capture de mejor forma la información que se busca. En este aspecto, la existencia de la herramienta que se propone podría ser un buen punto de partida para análisis de textos relacionados con opiniones, debates u otros.

Objetivos

Objetivo General

El objetivo general del trabajo de memoria es cuantificar la complejidad de un texto a través de la creación de una herramienta que reciba el texto en lenguaje natural y entregue la complejidad con la que éste se puede representar.

Objetivos Específicos

Para poder cumplir el objetivo general se proponen los siguientes objetivos específicos:

1. Desarrollar un método para cuantificar la complejidad de un texto en lenguaje natural, esto es, las dimensiones mínimas en que se puede representar.
2. Crear una herramienta que tenga como entrada texto en lenguaje natural y retorne la cuantificación de la complejidad del texto.
3. Obtener y recopilar datos de texto en lenguaje natural de opiniones, debates, noticias u otros; y ocuparlos para estudiar el uso de la herramienta y evaluar los resultados.

Capítulo 1

Estado del Arte

A continuación, se expondrán los conceptos más importantes para entender el estado actual del problema que se plantea y las herramientas para resolverlo. En la sección 1.1 se mencionará el trabajo que es el punto de partida de esta memoria. Luego se describirán recursos relacionados que serán utilizados como herramientas para encontrar la solución al problema planteado. Estos recursos son conocimientos claves para llegar a la solución propuesta en esta memoria.

Para los *frameworks* que serán descritos en esta sección (*SemAxis*, *FrameAxis* y *POLAR*) se realizarán descripciones de sus objetivos y metodologías para construir una solución. Si bien los objetivos de los trabajos mencionados difieren con el objetivo de este trabajo de título, sus metodologías servirán de base para la solución propuesta.

1.1. *Framework* supervisado para medir complejidad

La mayoría de los trabajos que buscan caracterizar las opiniones asumen que éstas se comporta como un número en la recta real. Esta unidimensionalidad es útil para entender opiniones desde dos perspectivas distintas (liberal o conservador, por ejemplo), pero potencialmente puede causar una reducción en la complejidad de la opinión, y por lo tanto, una representación pobre de opiniones multilaterales.

En el artículo titulado *On Complexity of Opinions and Online Discussions* [18] se presenta un *framework* que ocupa datos de foros de internet en el que las personas pueden votar los comentarios; dándoles su aprobación, desaprobación o simplemente no votarlos. Ocupando estos datos, se genera una matriz de signos alternos que asocia cada comentario a un vector de valoraciones donde cada dimensión es un usuario. Así, tomando en cuenta los comentarios de los participantes y sus opiniones expresadas a través de las valoraciones de los comentarios que hicieron de otras personas, el *framework* encuentra una representación de las opiniones en el mismo espacio latente. Luego, describe un set de algoritmos que sirven para estimar la cantidad de dimensiones mínimas que se necesitan para representar una opinión descrita en este espacio, y de esta manera, estimar la complejidad de las opiniones.

Al probar el mencionado modelo en datos obtenidos de foros de internet, se determinó que

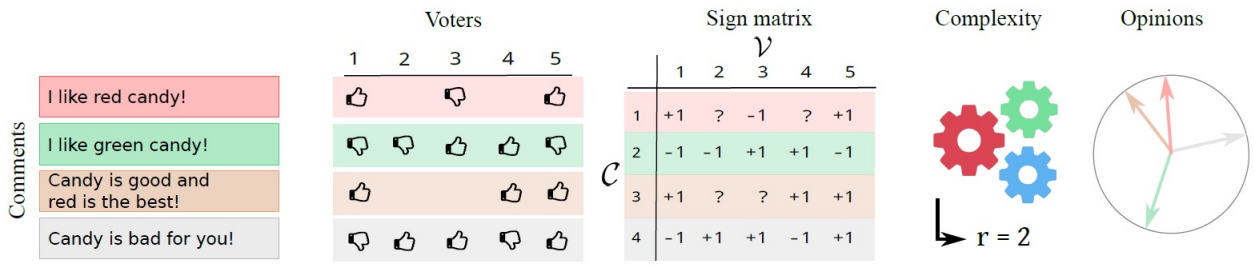


Figura 1.1: Imagen extraída del artículo sobre la complejidad de las discusiones que resume su metodología: Comienza con un conjunto de comentarios y de votaciones hechas por usuarios que son mapeados a una matriz de signos, luego se le aplican los cálculos de complejidad que permiten terminar con una nueva representación de las opiniones.

sólo el $\sim 25\,\%$ de las opiniones consideradas podían ser explicadas con la aproximación básica de una dimensión, el $\sim 60\,\%$ podían ser explicadas con una representación bidimensional y el resto con tres o más dimensiones [18].

Estos resultados demuestran la importancia de conocer la complejidad de las opiniones antes de intentar representarlas en un espacio dimensional en el que se puede perder información importante para el análisis que se busca; sin embargo, el *framework* mencionado tiene la limitante que necesita de las valoraciones de los participantes de la discusión para poder estimar la complejidad, por lo que es útil y novedoso este trabajo de memoria que se propone crear una herramienta que pueda cuantificar la complejidad de las opiniones sin tener valoraciones y solamente disponiendo del texto en lenguaje natural, ya que una herramienta de este tipo no existe hoy en día.

1.2. *Word Embeddings*

En este trabajo de título se utilizarán *word embeddings*, por lo que a continuación se presenta una breve introducción a este amplio tema.

Los *word embeddings* son representaciones vectoriales de palabras. Se utilizan para llevar las palabras, que son símbolos discretos, a un espacio vectorial en que se puede calcular la distancia o similitud entre palabras. Típicamente son vectores de números reales cuya dimensionalidad es dada. Frecuentemente la idea es que los *word embeddings* capturen información semántica de las palabras de tal manera que otras palabras que se encuentran cercanas en el espacio vectorial tengan significados similares.

Si bien existen distintas maneras de crear *word embeddings*, una de las técnicas más simples para entrenarlos en la práctica es *Word2Vec* [9] que usa un modelo de redes neuronales para aprender asociaciones de palabras de un conjunto de texto dado (denominados corpus).

Existen *word embeddings* preentrenados en grandes *datasets* que están disponibles para no tener que entrenar desde cero. Entre estos están: los *word embeddings* preentrenados *Word2Vec* de *Google* [1], que fueron entrenados en un *dataset* de noticias de *Google* con cerca

[1] `https://code.google.com/archive/p/word2vec/`

de cien billones de palabras; los *word embeddings* preentrenados *GloVe* [14] de la universidad de *Standford*, que tiene opciones de *embeddings* entrenados en *Twitter* o Wikipedia [2].

Es importante mencionar que las dimensiones de los *word embeddings* no son directamente interpretables ya que representan características latentes de las palabras.

1.3. *Frameworks* que utilizan diferenciales semánticos

1.3.1. *SemAxis*: Caracterización semántica

SemAxis [1] es un *framework* que busca caracterizar la semántica de las palabras para dominios específicos. Parte de la base que las palabras tienen distintos significados en contextos diferentes, por ejemplo, la palabra suave puede tener una connotación positiva al hablar de animales de peluche, pero negativa al hablar de deportes como el hockey.

Esto difiere de la suposición que se hace tradicionalmente en un análisis de texto basado en *lexicon* (diccionarios de palabras que se asocian a un sentimiento), donde se asume que las palabras no cambian significativamente su significado en diferentes contextos. Esta limitación ha sido reconocida y existen recursos que crean *lexicon* de sentimientos para dominios específicos [5]. Pero SemAxis pretende caracterizar la semántica de las palabras en dominios específicos más allá de los sentimientos.

Para caracterizar la variedad semántica de las palabras de manera sistemática *SemAxis* propone utilizar los ejes semánticos. Los ejes semánticos (o diferenciales semánticos) representan la diferencia que existe entre dos palabras, típicamente se escogen dos palabras antónimas para que el eje represente la diferencia entre dos conceptos polares. Los ejes propuestos son 732 que provienen de pares de antónimos que provee *ConceptNet* [17].

Para generar los ejes semánticos se necesita representar las palabras de manera vectorial, para eso utilizan *word embeddings*. La metodología utilizada para crear un eje semántico consiste en primero, escoger dos palabras antónimas, y para cada una obtener las l palabras cuyos *word embeddings* son más similares. A continuación, se agrupan las palabras antónimas y sus l palabras más similares en conjuntos de $l+1$ palabras. Luego, se calcula el vector promedio entre los *word embeddings* para cada uno de los dos conjuntos de palabras. Finalmente, se restan los dos vectores promedios de cada conjunto, que representan dos polos antónimos cada uno, y el vector resultante es lo que se considera el eje semántico.

Luego de obtener el vector que representa al eje semántico, se puede calcular la similitud de una palabra a ese eje semántico. En *SemAxis* se escoge utilizar coseno similitud para esta labor, a esto le denominan *score* que, para un vector de un eje semántico V_{axis} y una palabra w con un vector que la representa v_w, se define así:

$$score(w)_{V_{axis}} = cos(v_w, V_{axis}) = \frac{v_w \cdot V_{axis}}{||v_w||\,||V_{axis}||} \tag{1.1}$$

[2] `https://nlp.stanford.edu/projects/glove/`

El resultado de la similitud coseno indica qué tan alineada está la palabra con el eje semántico. De tal manera que un *score* más alto representa una alineación más alta al polo que fue restado (minuendo), y un *score* más bajo representa una alineación mayor al polo restante (sustraendo).

Para construir los *word embeddings* utilizan el modelo *Word2Vec* y entrenan los *word embeddings* en el corpus que quieran analizar. Pero para corpus más pequeños, utilizan *word embeddings* entrenados en *datasets* grandes y luego entrenados en el corpus pequeño que se quiere analizar.

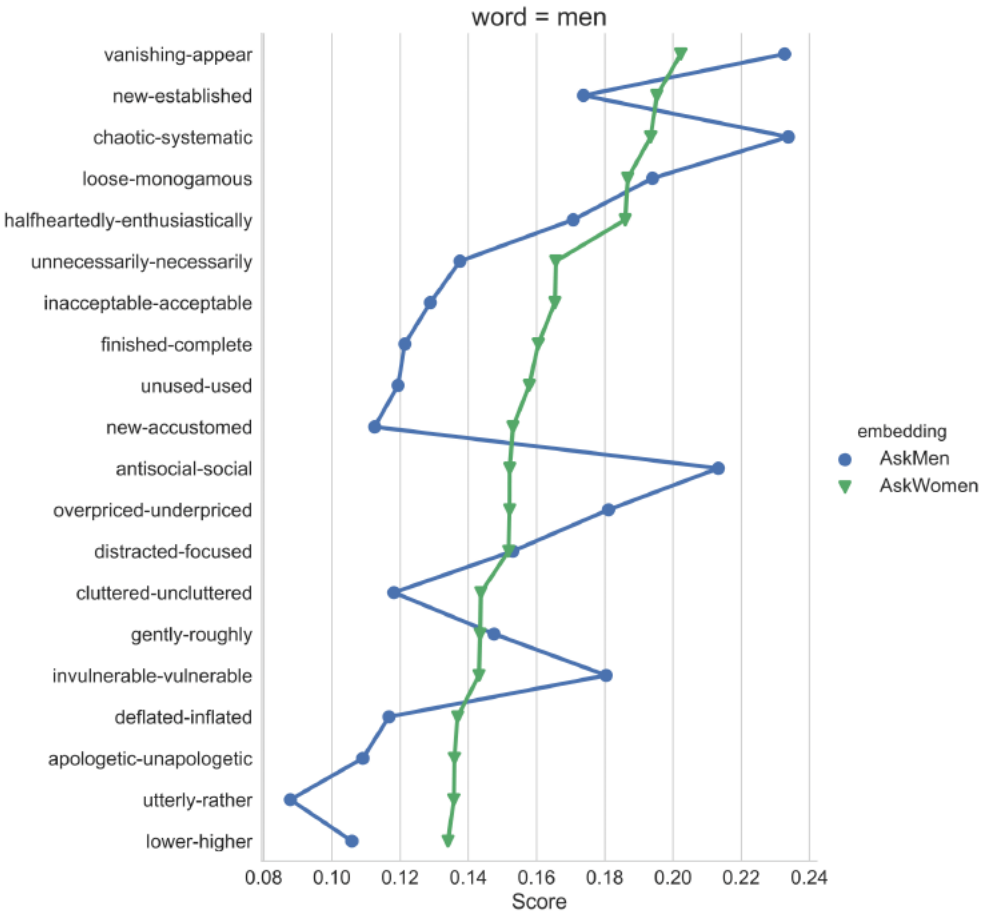


Figura 1.2: Imagen extraída del artículo de *SemAxis* que ejemplifica los resultados obtenidos para la palabra *men* en los foros de *Reddit AskWomen* y *AskMen*.

1.3.2. *FrameAxis*: Caracterización para sesgos

FrameAxis [6] es un *framework* que tiene el objetivo de caracterizar un texto con atributos relacionados al *framing*. El *framing* es una forma de influir en la percepción de una idea a través de una acentuación sin la necesidad de hacer un argumento sesgado [3]. Esto lo hacen identificando los ejes semánticos que están más sobrerrepresentados en un texto.

A las palabras antónimas que generan un eje semántico le llaman *microframe*, por ejemplo legal-ilegal, justo-injusto, etc. Y las métricas que proponen son *microframe bias*, que captura

cuán sesgado está un texto con respecto a cierto *microframe*, y *microframe intensity*, que muestra qué tan activamente se está usando cierto *microframe*.

Para representar las palabras vectorialmente usan los *word embeddings* preentrenados de GloVe y para generar los *microframe* usan 1828 pares de antónimos provenientes de WordNet [10], de los cuales descartan 207 que no se encuentran entre los *word embeddings*, quedando un total de 1621 pares de antónimos.

En *FrameAxis* los ejes semánticos no se calculan en base a conjuntos de las palabras más similares a las palabras antónimas, simplemente se restan los vectores de las palabras antónimas.

Las fórmulas propuestas para *microframe bias* y *microframe intensity* tienen como base lo que *SemAxis* denomina como *score*, pero en este *framework* se le llama contribución. Es decir, la contribución de una palabra a un *microframe* es el coseno similitud entre el vector de la palabra y el vector que representa el eje semántico del *microframe*.

Entonces, se define *microframe bias* del corpus t con respecto al *microframe* f como:

$$B_f^t = \frac{\sum_{w \in t}(n_w c_f^w)}{\sum_{w \in t} n_w} \tag{1.2}$$

Donde n_w es la frecuencia de la palabra w en el corpus t y c_f^w es la contribución de la palabra w con respecto al *microframe* f.

Mientras que el *microframe intensity* para un corpus t con respecto a un *microframe* f se define como:

$$I_f^t = \frac{\sum_{w \in t} n_w (c_f^w - B_f^T)^2}{\sum_{w \in t} n_w} \tag{1.3}$$

Donde B_f^T es el *microframe bias* calculado para todo el corpus, n_w es la frecuencia de la palabra w en el corpus t y c_f^w es la contribución de la palabra w con respecto al *microframe* f. *Microframe intensity* muestra qué tan intensamente se usa un *microframe* en un corpus calculando la varianza de la contribución de la palabra en un *microframe* para todas las palabras de dicho corpus.

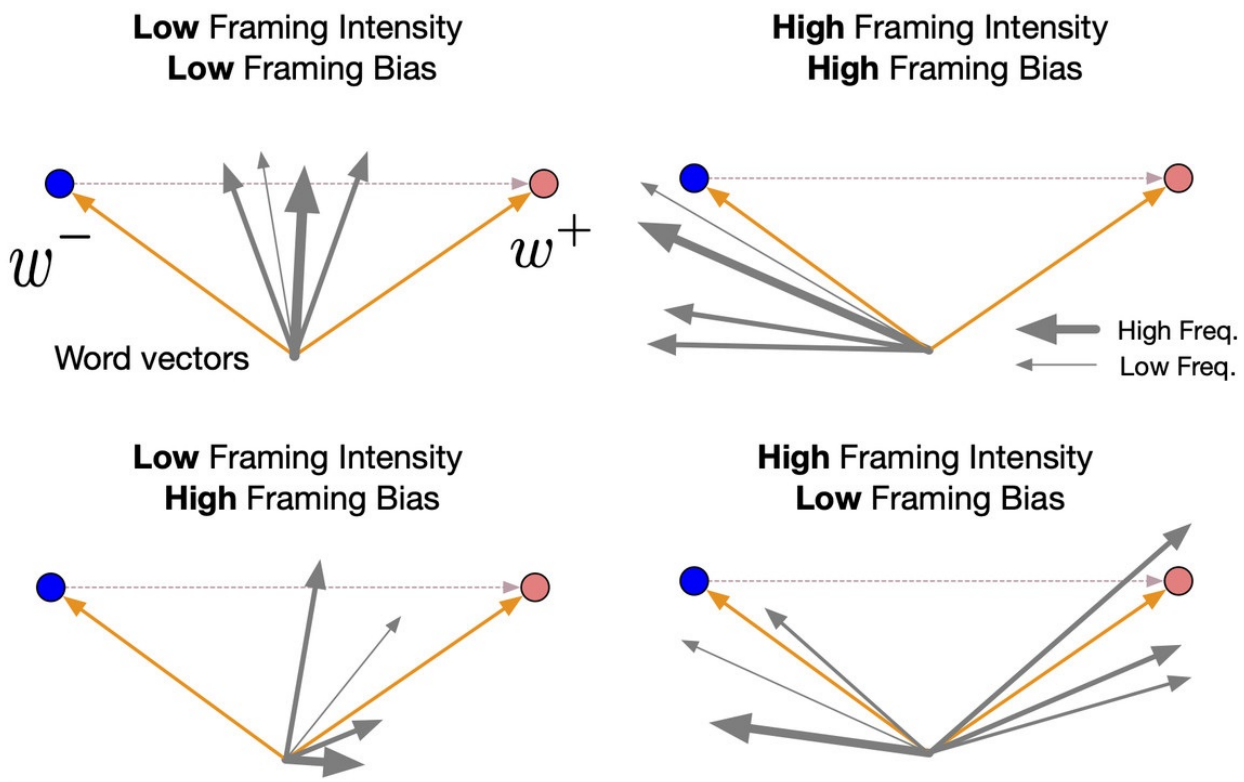


Figura 1.3: Ilustración extraída del artículo de *FrameAxis* [6] en que se aprecia gráficamente el significado de los conceptos de *microframe bias* y *microframe intensity*.

1.3.3. *POLAR*: Interpretabilidad a *word embeddings*

POLAR [8] es un *framework* cuyo objetivo es agregarle interpretabilidad a *word embeddings* preentrenados con el uso de diferenciales semánticos que es lo mismo que llaman ejes semánticos en *SemAxis* o *FrameAxis*. Como resultado final se obtienen nuevos *embeddings* interpretables.

En *POLAR* usan *word embeddings* preentrenados de GloVe y Word2Vec. Y para generar los ejes semánticos utilizan palabras obtenidas de WordNet.

La idea es transformar los *word embeddings* preentrenados, usando los ejes semánticos de palabras polares (extienden el uso de palabras antónimas), a un nuevo espacio polar con dimensiones interpretables definidas por las palabras polares utilizadas. Esto se hace identificando un subespacio interpretable y proyectando los *word embeddings* preentrenados a este subespacio, obteniendo dimensiones interpretables que denominan dimensiones polares.

Formalmente, para un conjunto de N pares de palabras polares P:

$$P = \{(p^1_{+z}, p^1_{-z}), (p^2_{+z}, p^2_{-z}), ..., (p^N_{+z}, p^N_{-z})\} \tag{1.4}$$

Por ejemplo, el par p^1 podría representar al *microframe* caliente-frío, siendo $+z$ la palabra

caliente y $-z$ la palabra frío. Cabe mencionar que la elección de la palabra positiva y la palabra negativa es arbitraria.

Luego, para un par n perteneciente a P se puede obtener la dirección dir_n:

$$\mathrm{dir}_n = V_{p^n_{+z}} - V_{p^n_{-z}} \tag{1.5}$$

Donde $V_{p^n_{\pm z}}$ es el vector (*word embedding*) que representa a la palabra $\pm z$ del par p^n

Obteniendo la dirección para todos los pares de P se obtiene la matriz $\mathrm{dir} \in \mathbb{R}^{N \times \mathrm{d}}$, donde d es la dimensión de los *word embedding* (en *POLAR* se utilizaron *word embedding* de 300 dimensiones).

Para encontrar las dimensiones polares se realiza el siguiente cálculo, sea una palabra w con un vector que la representa V_w, se denota el subespacio E_v tal que:

$$\mathrm{dir}^T E_v = V_w \tag{1.6}$$

$$E_v = (\mathrm{dir}^T)^{-1} V_w \tag{1.7}$$

Se observa que se necesita trasponer la matriz de direcciones para hacer calzar las dimensiones y que por la composición del nuevo espacio E_v cada dimensión puede ser interpretada en términos de palabras polares (como frio-caliente, legal-ilegal, republicano-demócrata, etc.).

Además, como la matriz de direcciones no necesariamente es invertible en *POLAR* se utiliza la inversa generalizada de Moore-Penrose.

Realizar este cálculo para todas las palabras que están en los *word embeddings* significa crear nuevos *word embeddings* interpretables. La cantidad de dimensiones que tendrán estos *embeddings* son a lo más N y son dadas.

En el caso que se quiera elegir un número arbitrario de dimensiones menores a N, en *POLAR* proponen tres formas de elegir k dimensiones entre N dimensiones disponibles.

- Aleatorio: Elegir aleatoriamente k dimensiones dentro de las N dimensiones.
- Maximización de varianza: Se calcula el valor de cada palabra del vocabulario cuando es mapeado a la dimensión (eje semántico) y luego la varianza de estos valores para todas las dimensiones. Se escogen las k dimensiones con mayor varianza.
- Maximización ortogonal: Primero, se escoge la dimensión con mayor varianza de la misma manera que el método anterior, y luego se seleccionan $k-1$ dimensiones de tal manera que cada dimensión que se agrega tiene la mayor ortogonalidad con respecto a las dimensiones ya seleccionadas. Por lo tanto, un vector candidato z para el conjunto O de vectores, que representan las dimensiones seleccionadas del conjunto de todas las dimensiones M, se puede describir de la siguiente manera:

$$z = \underset{x \in M}{\operatorname{argmin}} \frac{1}{|O|} \sum_{n=1}^{n=|O|} \vec{O}_{\mathrm{i}} \cdot \vec{x} \tag{1.8}$$

Luego de obtener la dimensión con mayor varianza, se saca ésta del conjunto M y se realiza la búsqueda de la dimensión z que maximice ortogonalidad. En cada paso se saca la dimensión escogida del conjunto M para no repetir la dimensión.

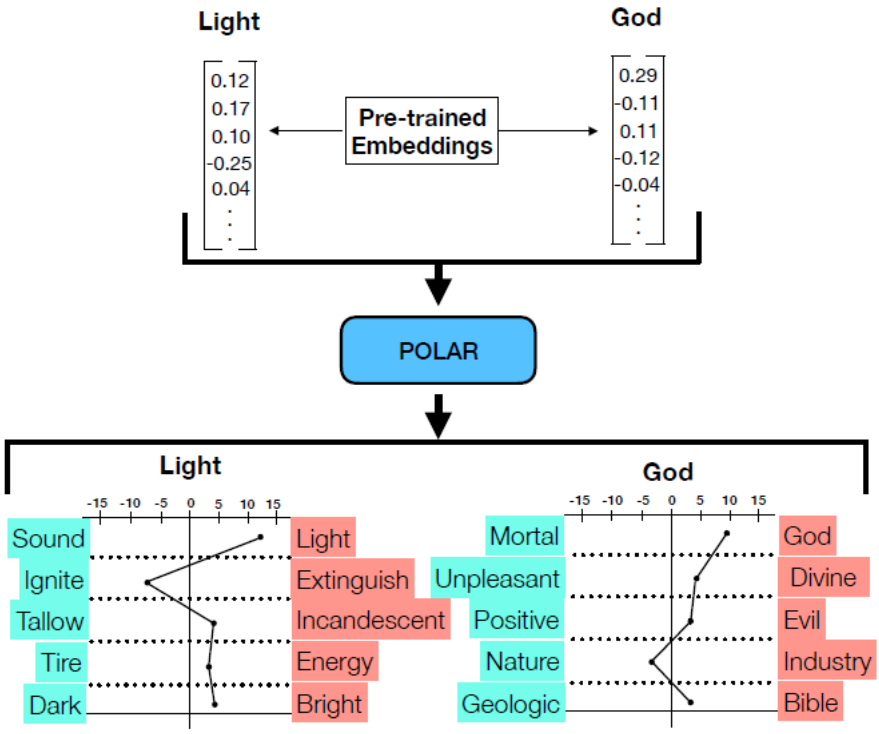


Figura 1.4: Imagen extraída del artículo de *POLAR* [8] que ejemplifica la transformación de *word embeddings* preentrenados a *embeddings* polares.

1.4. Análisis de componentes principales

El análisis de componentes principales es una técnica que comúnmente se utiliza para reducir la dimensionalidad de un conjunto de datos [12]. Para esto proyectan los datos originales a nuevos ejes llamados componentes principales, los que son ortogonales entre sí y que cada uno representa un porcentaje de la varianza original. Con esto se tienen tantos componentes principales como dimensiones originales y, además, se tienen ordenandos por la cantidad de varianza original que representan. Luego, si el objetivo es obtener una representación de los datos originales en menos dimensiones, simplemente se toman tantos componentes principales como nuevas dimensiones se requieran, eligiendo los componentes principales que mayor cantidad de varianza representan.

Uno de los usos más comunes del análisis de componentes principales es cuando se quieren graficar en dos o tres dimensiones datos que tienen cuatro o más dimensiones. De esta

manera, se realiza el análisis de componentes principales y se eligen las primeras dos o tres componentes principales para mapear los datos originales y luego graficarlos.

Otra alternativa a la elección de la cantidad componentes principales que se seleccionan es determinar cuánta varianza se quiere representar de los datos originales y luego tomar los componentes principales que juntos representen esa varianza.

Por definición el primer componente principal es el que tiene más varianza (porque el procedimiento para generar los componentes principales escoge el primero maximizando la varianza que se puede representar en un nuevo eje a partir de los datos originales) y luego los demás componentes principales son dimensiones ortogonales al primer componente, que luego son ordenados según la cantidad de varianza que representan.

Además, los componentes principales son combinaciones lineales de las dimensiones originales, por lo que se pueden escribir en función de éstas; por lo tanto, combinando las dimensiones originales y asociándoles un peso, se puede construir cada uno de los componentes principales. Esto es útil ya que se puede interpretar como qué tan influyentes son las dimensiones originales en la composición de un componente principal. Usualmente, este análisis se realiza con el primer componente principal al ser el que representa más varianza.

Capítulo 2

Desarrollo de la Solución

En este capítulo se describirán los supuestos provenientes de la lingüística que se hicieron para modelar la solución aprovechando los *frameworks* presentados en el capítulo anterior. Luego se mostrará cómo se implementó ese planteamiento en el lenguaje *Python*.

2.1. Modelamiento de la Solución

La técnica de los diferenciales semánticos de Osgood [11] supone que un concepto tiene diversas propiedades asociadas a él, como por ejemplo si es veloz o lento, o si es bueno o malo, entre otras; y se usa para medir el significado connotativo que las personas les dan a las palabras, a través de la valoración dentro de un espectro de esas propiedades o dimensiones que se representan con palabras polares. Como la composición de las dimensiones mencionadas se puede hacer con palabras que representan los dos polos de un concepto, esta solución se basa en el supuesto que esas dimensiones pueden ser utilizadas para comprender las dimensiones en las que se caracteriza una discusión o debate. De esta manera, generar los *embeddings* polares con el *framework* de *POLAR* (mencionado en la sección 1.3.3), a partir de *word embeddings*, es generar una valoración cuantitativa de una palabra con respecto a cada una de las dimensiones que se consideren.

Los diferenciales semánticos se basan en las valoraciones que las personas hacen de las palabras respecto a las propiedades que tienen asociadas, por lo que su utilización para encontrar la complejidad de las dimensiones en opiniones y debates mantiene la noción de connotatividad en el significado de las palabras. Así mismo, utilizar *word embeddings* dentro de la solución también mantiene la misma noción porque estos intentan capturar el significado de las palabras a través del uso que se les dan. Sin embargo, extender estos supuestos a todo tipo de textos en lenguaje natural tiene sentido en la medida que se asuma algún grado de subjetividad en estos.

Un paso importante que considerar cuando se generan *embeddings* polares son los pares polares que se usarán para representar los diferenciales semánticos (las dimensiones del texto). En los *frameworks* Polar, SemAxis y FrameAxis se utilizan pares de antónimos para generar los diferenciales semánticos, pues al ser opuestos se espera que representen un espectro amplio

como dimensión. Pero no hay limitación en confeccionar las dimensiones con palabras que no son opuestos en términos denotativos, pero que sí lo son dentro de un contexto (por ejemplo, considerar como par polar las palabras republicano y demócrata). Dado lo anterior se puede considerar que los pares polares confeccionados para un contexto particular pueden aprovechar mejor dicho contexto, pero utilizar pares de antónimos entrega más generalidad para comparar distintos contextos. No obstante, es importante mencionar que crear una lista de pares polares para cada contexto puede ser una tarea lenta y sensible al sesgo si se realiza manualmente.

Gracias a *POLAR* podemos representar cada palabra con un *embedding* polar, y dado que un texto o documento está compuesto por un conjunto de palabras, se puede representar el documento como una matriz de *embeddings* polares. De esta manera, se tiene una representación de la opinión o documento en un espacio, que es un paso previo para cumplir el objetivo de la cuantificación de complejidad. A continuación, se tiene una matriz con tantas filas como palabras tenga el texto y tantas columnas como pares polares se consideren. Dado el supuesto en el que se basa esta solución, se sabe que muchas palabras pueden tener valoraciones cercanas al nulo en varias dimensiones. Y si se analiza el conjunto completo de palabras que representan un texto con respecto a cada dimensión, se puede tener que algunas dimensiones tienen pocas valoraciones significativas a lo largo del documento (una valoración significativa puede ser positiva o negativa, dependiendo a cuál de los polos se incline dentro del espectro que representa la dimensión). Tomando en cuenta que la motivación es encontrar la cantidad mínima de dimensiones en que se pueden representar las opiniones, esto se puede interpretar como la cantidad mínima de dimensiones que se necesitan para representar los documentos vistos como matrices de *embeddings* polares. Luego, se puede utilizar el método de análisis de componentes principales, mencionado en la sección 1.4, para estimar cuántas dimensiones se necesitan para realizar un cambio de eje de coordenadas que represente cierto porcentaje de la varianza. Notar que no se utiliza el análisis de componentes principales para llevar los datos originales a una dimensionalidad dada, si no que se decide qué porcentaje de varianza se quiere representar. Esto último puede servir como una medida de sensibilidad, ya que los diferenciales semánticos indican que las palabras tienen significado connotativo en diferentes dimensiones y en diferentes valoraciones, por lo tanto, que hay dimensiones que se valoran más que otras dentro de un conjunto de palabras. Lo anterior es diferente al punto de partida del trabajo de memoria, el *framework* que estima la complejidad a través de los patrones de votación, ya que éste no considera que pueden haber varias dimensiones participando con diversa relevancia.

Es importante destacar que tener la representación de los textos en *embeddings* polares concede interpretabilidad a las dimensiones, una característica no considerada en los objetivos de este trabajo, ya que no existe en el *framework* que es el punto de partida para esta memoria, que sólo reconoce patrones de votaciones. Pero con el método presentado no es directa la interpretación de las dimensiones dada la cuantificación de la complejidad de un texto, porque el análisis de componentes principales entrega nuevos ejes de coordenadas donde las dimensiones obtenidas son una combinación lineal de las dimensiones originales, por lo que para cada componente principal se tienen asociados pesos de las dimensiones originales, que en este caso están compuestas a partir de pares de palabras polares. A partir de estos pesos se puede saber qué dimensiones aportan más para la composición de un componente principal, lo que puede ser importante en los componentes principales que representan mayor varianza,

en particular el primero. Este análisis se puede acompañar de los métodos *microframe bias* y *microframe intensity* aportados por *FrameAxis* 1.3.2, pero excede a los objetivos planteados para esta memoria.

Con el método descrito se puede obtener una medida de complejidad para un documento, pero si se considera un debate con muchas opiniones y se quiere saber la complejidad del debate, es decir, la complejidad de un conjunto de opiniones o documentos, se puede usar el mismo método si se tiene una representación vectorial para cada documento. De esta manera, tal como para calcular la complejidad de una opinión, que se modela con una matriz de polar *embeddings* que representan las palabras, se requiere una matriz de vectores donde cada vector sea un *embedding* polar que represente los documentos. Esto se puede lograr modelando cada documento como el vector promedio de los *embeddings* polares que lo forman, luego el procedimiento usando el análisis de componentes principales es idéntico.

2.2. Implementación de la Solución

La solución descrita en este capítulo está implementada en lenguaje *Python* porque tiene facilidades como la variedad de librerías que se utilizan en esta implementación y la experiencia previa en este lenguaje. Igualmente, se emplea el inglés para documentar y dar nombre a las variables en el código, por ser un idioma de común uso en la computación.

El código correspondiente a la implementación se encuentra en el repositorio de *Github* anexo en este informe [1]. Dentro del código se encuentra el archivo DimCuantifier.py, que tiene la clase *DimCuantifier* donde está la parte más esencial del desarrollo de la solución y el archivo PreProcessingDimCuantifier.py que contiene la clase *PreProcessingDimCuantifier* que aporta distintos métodos útiles para hacer uso de *DimCuantifier*. Además, el repositorio consta con una serie de archivos en formato *Jupyter Notebook* que presentan diferentes casos de estudios, evaluaciones y un tutorial. Entre estos está el archivo dim_cuantifier_tutorial.ipynb que es un tutorial que ejemplifica el uso básico de la clase *DimCuantifier*.

2.2.1. Clase para el preprocesamiento

En la clase *PreProcessingDimCuantifier* está el método *preprocess_document* que recibe un documento y un modelo (típicamente es un modelo de *word embeddings*, estos funcionan como un diccionario que posee como llaves las palabras y como valores los vectores que representan las palabras) y devuelve una lista con el documento *tokenizado* (es decir, con cada palabra del documento siendo un elemento de la lista) y filtrado de tal manera que descarta las palabras del documento que no se encuentran en el modelo, esto es útil cuando se utilizan *word embeddings* pre-entrenados, ya que pueden existir palabras dentro del documento que no hayan sido consideradas en el corpus en el que se pre-entrenaron los *word embeddings*. También filtra las palabras que se encuentran en el conjunto de *stop words*, que son palabras tan comunes que por lo regular son descartadas en los modelos de Procesamiento de Lenguaje Natural para que no interfieran como ruido dentro de lo que realmente se quiere capturar con los *word embeddings*, que son características semánticas de las palabras. Las palabras del conjunto de *stop words* son obtenidas de la librería *nltk*, por defecto se consideran las del

[1] `https://github.com/IgnacioDL/DimCuantifier`

idioma inglés, pero *PreProcessingDimCuantifier* tiene el método *change_ stopwords_ language* para cambiar de lenguaje dentro de los soportados por *nltk* [2]. Además, se consideran símbolos de puntuación dentro del conjunto de las *stop words*.

La clase incluye pares de palabras que provienen del listado de antónimos de *WordNet* para cargar como pares polares con el método *load_ polar_ words_ wordnet*. *WordNet* es la misma fuente que utiliza *POLAR*. De lo anterior se obtiene una lista de tuplas, donde cada tupla representa un par polar y dentro del par cada palabra está en formato *string*, todo este formato es útil para utilizar como *input* en la clase *DimCuantifier*. Además, con el método *select_ polar_ words_ list* se entrega la opción de cargar una lista de tuplas de palabras, pero donde pueden haber palabras repetidas, en dados casos el método selecciona el par que tenga menor coseno similitud, ya que la idea es considerar el par que tiene un espectro más grande en su diferencial semántico.

El último método que tiene la clase es *generate_ norm_ embedding* que recibe un modelo de *word embeddings*, normaliza sus vectores y lo guarda en una ruta señalada.

2.2.2. Clase con el núcleo de la solución: *DimCuantifier*

La clase *DimCuantifier* es la que tiene implementada el núcleo de la solución. Para crear el objeto de la clase se requiere como *input* un modelo de *word embeddings* y una lista de tuplas con los pares polares a utilizar. Apenas se crea el objeto se genera un listado de vectores que representan los diferenciales semánticos en el mismo orden de la lista de pares polares entregada. Esto quiere decir que se utilizan los vectores del modelo de *word embeddings* entregado para restar los vectores de las palabras en cada par polar.

Para comenzar a usar la clase es necesario agregar un corpus, esto se realiza con el método *set_ corpus* o con *add_ to_ corpus*, en caso de que se quiera agregar más documentos al corpus existente. Los requerimientos de formato para el corpus ingresado es que sea una lista de documentos, y que cada documento sea una lista de palabras en formato *string*, es decir, una lista de documentos *tokenizados*. Una vez agregado el corpus se crea un diccionario de frecuencia de las palabras presentes en el corpus completo.

Los métodos *generate_ word_ polar_ dimensions*, *generate_ document_ polar_ dimensions* y *generate_ corpus_ polar_ dimensions* sirven para obtener representaciones vectoriales en *embeddings* polares de las palabras, documentos y el corpus respectivamente. En el caso de las palabras se genera un diccionario en que las llaves son las palabras y los valores los vectores polares. Este diccionario es beneficioso si se quiere utilizar como modelo de *word embeddings*. En cambio, para los documentos y el corpus se genera una representación vectorial para cada uno. El procedimiento para producir estas representaciones polares consiste, en el caso de las palabras, en multiplicar el vector de cada palabra (propiciado por el modelo de *word embeddings*) con el resultado de invertir la matriz transpuesta de los vectores que representan los diferenciales semánticos, tal como se describe en *POLAR*. Para invertir la matriz se utiliza la función *pinv* (abreviatura de *pseudo-inverse*) de la sección de álgebra lineal de la librería *numpy* que ejecuta la pseudoinversa de Moore-Penrose. Con respecto a los documentos el procedimiento primero promedia los vectores de las palabras que aparecen en el documento

[2] `https://www.nltk.org/`

y luego realiza la multiplicación del vector con la matriz de diferenciales semánticos (transpuesta e invertida). Mientras que es análogo para el corpus, que primero debe representarse como el vector promedio de los vectores que representan a los documentos.

Para cuantificar la dimensión de los documentos o del corpus completo existen las funciones *cuantify_dim_documents* y *cuantify_dim_corpus* respectivamente. El *output* de ambas funciones es un número entero que indica cuántos componentes principales se requieren para representar la varianza establecida. Por defecto el porcentaje de varianza que se busa representar es de 99 %, pero puede ser cambiado con la función *set_percentage_variation*. Con tal de realizar el análisis de componentes principales se utiliza la clase *PCA* (sigla para *Principal Component Analysis*) de la librería *scikit-learn* [13]. Dentro de las funciones cuantificadores se utiliza la función *StandardScaler* de la sección de *preprocessing* también de la librería *scikit-learn*. La razón es que es importante asegurar que los datos estén centrados y escalados para realizar un análisis de componentes principales, pues la optimización que se realiza para encontrar el primer componente principal intenta maximizar la varianza en los ejes candidatos que son generados con respecto al origen, por lo tanto, un origen recentrado permite a los ejes candidatos ajustarse mejor a la distribución de los datos, y la normalización de los datos permite no sesgar el análisis de componentes principales a una dimensión con magnitudes muy altas.

Cuantificar las dimensiones del corpus requiere previamente generar las dimensiones polares que representan los documentos (con el método *generate_document_polar_dimensions* se hace internamente) y luego aplicar el análisis de componentes principales con un objeto de la clase *PCA*. Así mismo, para cuantificar las dimensiones de cada documento, se requiere tener una representación en dimensiones polares de las palabras (se hace internamente con el método *generate_word_polar_dimensions*) y se necesita un objeto de la clase *PCA* para cada uno de los documentos; todos los objetos de la clase *PCA* se almacenan en una lista con el mismo orden que los documentos.

Una vez cuantificadas las dimensiones de los documentos o el corpus, se pueden obtener los pesos que tienen las dimensiones originales (que son dimensiones polares) en los componentes principales. Para esto existen las funciones *calculate_loading_scores_corpus* y *calculate_loading_scores_documents* para obtener *DataFrames* de la librería *pandas* de los corpus y documentos respectivamente. En el caso de los documentos se genera una lista de *DataFrames* con los resultados para cada documento con el mismo orden de los documentos. Estos *DataFrames* tienen como índice los ejes semánticos y como columna los componentes principales.

También se ofrece en *DimCuantifier* las funciones planteadas por *SemAxis* y *FrameAxis*: contribución, *Intensity* y *Bias* con los métodos *contribution* y *calculate_bias_and_intensity*. En el caso de la contribución simplemente se realiza un cálculo de coseno similitud que es útil para implementar *Bias* e *Intensity*. El método *calculate_bias_and_intensity* genera un *DataFrame* donde los índices son los ejes semánticos y las columnas son los resultados del cálculo de *Bias* e *Intensity* sobre el corpus. Y tal como se ofrece en *POLAR* se puede seleccionar una cantidad arbitraría de dimensiones para calcular las dimensiones polares, por defecto son 10 dimensiones arbitrarías, pero pueden ser modificadas con el método *set_n_arbitrary_dimensions*. Esta selección de dimensiones se efectúa descartando las di-

mensiones con menor *Intensity* con el método *select_dim_by_intensity*.

Capítulo 3

Casos de Estudio

En este capítulo se estudia la solución desarrollada en un caso práctico con la intención de validar y ajustar el método propuesto en datos de la vida real.

3.1. Noticias de la *BBC*

En este caso de estudio se utiliza un *dataset* de noticias de la cadena *BBC* (*British Broadcasting Corporation*). El conjunto de datos fue obtenido de *kaggle* [1] [4] y consiste en 2225 documentos provenientes de la página web de *BBC News* publicados entre 2004 y 2005. Estas noticias pertenecen a las categorías: Política, Deporte, Tecnología, Entretenimiento o Negocios. Además de la categoría, el *dataset* incluye título y contenido para cada documento, sin elementos nulos.

En la tabla 3.1 se muestran la cantidad de noticias por categoría y en el gráfico 3.1 se ilustran estas cantidades en base a la porción que representan.

Categoría	Número de Noticias
Deportes	511
Negocios	510
Política	417
Tecnología	401
Entretenimiento	386

Tabla 3.1: Cantidad de noticias por categoría en el *dataset* de noticias de la *BBC*

[1] `http://mlg.ucd.ie/datasets/bbc.html`

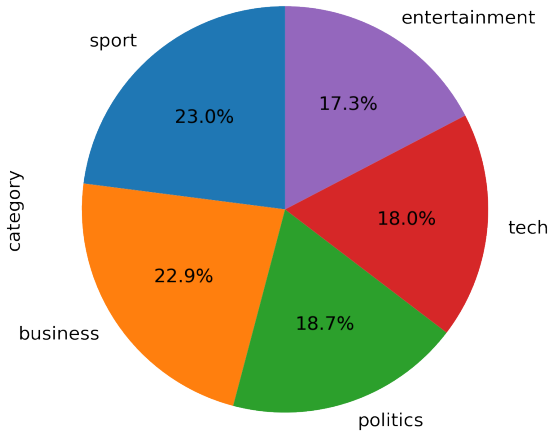


Figura 3.1: Porcentajes de noticias por categoría en el *dataset* de noticias de la *BBC*.

3.1.1. Preprocesamiento e *input* de *DimCuantifier*

Se empleó la clase *PreProcessingDimCuantifier* para *tokenizar* el contenido de los documentos del *dataset*.

Durante el estudio del *dataset* de noticias de la *BBC* se utilizó como *input* para la clase *DimCuantifier*: un modelo de datos de *word embeddings* preentrenados de GloVe con 42 billones de palabras, cargado con la librería *gensim* [15] y su método *Word2Vec* [9]; y el otro input fue una lista de pares polares constituida por antónimos extraídos de la fuente especializada *WordNet* (tal como se realiza en *POLAR*), que se compone de 1468 pares polares luego de filtrar palabras repetidas con el método descrito en la sección 2.2.1. Ambas variables se mantuvieron inalteradas durante todo este caso de estudio.

Luego se estableció como corpus el contenido *tokenizado* de los documentos del *dataset*.

3.1.2. Correlación entre dimensiones estimadas y tamaño

En esta sección se describirá el análisis de la relación entre el tamaño de los documentos (cuántas palabras tiene) y la cantidad de dimensiones que son cuantificadas con el método propuesto, y cómo los resultados obtenidos llevan a concluir que la mejor opción para comparar documentos con este método es en condiciones de tamaño similares. Para realizar esto se obtuvieron las dimensiones estimadas para cada documento considerando la reducción dimensional con el método de análisis de componentes al 99 %, 75 %, 50 % y 25 % de varianza representada. Estos valores fueron escogidos para notar de manera notoria el efecto de la diferencia de varianza. En este proceso también se cuantificó las dimensionalidades de los documentos reduciendo *word embeddings* y *embeddings* polares, para comparar las diferencias entre reducir de una manera u otra.

A continuación, se pueden ver en la figura 3.2 un mapa de calor con los resultados del cálculo de correlación entre las dimensiones cuantificadas, a distintos niveles de varianza, y el

largo de su contenido *tokenizado*, realizado con el método de *pearson* dispuesto por la librería *pandas*.

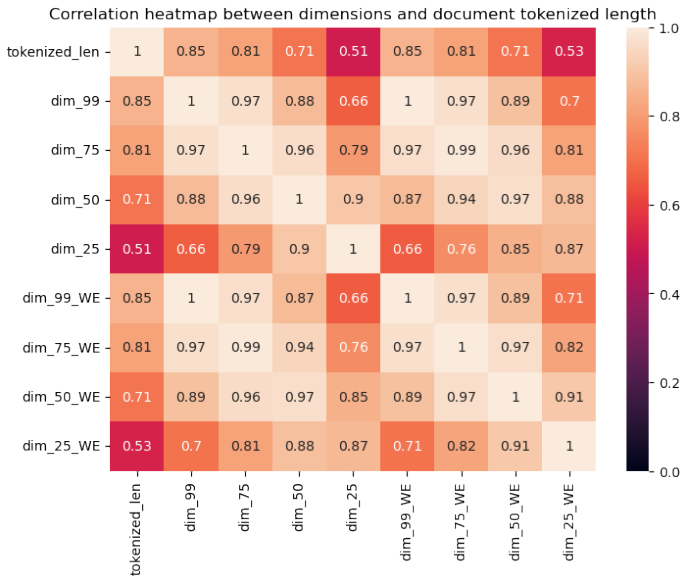


Figura 3.2: En la figura se puede ver la correlación entre el largo del contenido *tokenizado* de un documento y sus dimensiones cuantificadas a diferentes niveles de varianza, donde las etiquetas con el sufijo *WE* indican que la reducción se hizo sobre *word embeddings*, de lo contrario se hizo sobre *embeddings* polares.

El mapa de calor 3.2 muestra que la correlación entre la cantidad de palabras del documento y las dimensiones cuantificadas es positiva y alta, pero que disminuye a medida que baja la varianza representada en el análisis de componentes principales. La correlación entre las dimensiones cuantificadas sigue la misma lógica. También se puede apreciar que la correlación entre las dimensiones obtenidas reduciendo sobre *word embeddings* y *embeddings* polares son muy cercanas al 1 para la misma varianza representada.

La dependencia tan alta del tamaño del documento no es una característica deseada por el sesgo que implica, por lo que se replicó el análisis reduciendo el tamaño de los documentos más largos. Para esto, se eligió la mediana de la cantidad de palabras de los documentos del *dataset* como límite de largo, de tal manera que los documentos cuyo largo sea mayor a la mediana son acortados para tener el mismo largo que la mediana. La mediana mencionada es de 132, truncada para que sea un número entero. Dicho de otra forma, se tomaron las primeras palabras de los documentos a modo que el largo de cada documento sea igual o menor a la mediana de largos del *dataset*. Las nuevas dimensiones con los documentos acortados generan el siguiente mapa de calor de correlaciones que se ve la figura 3.3.

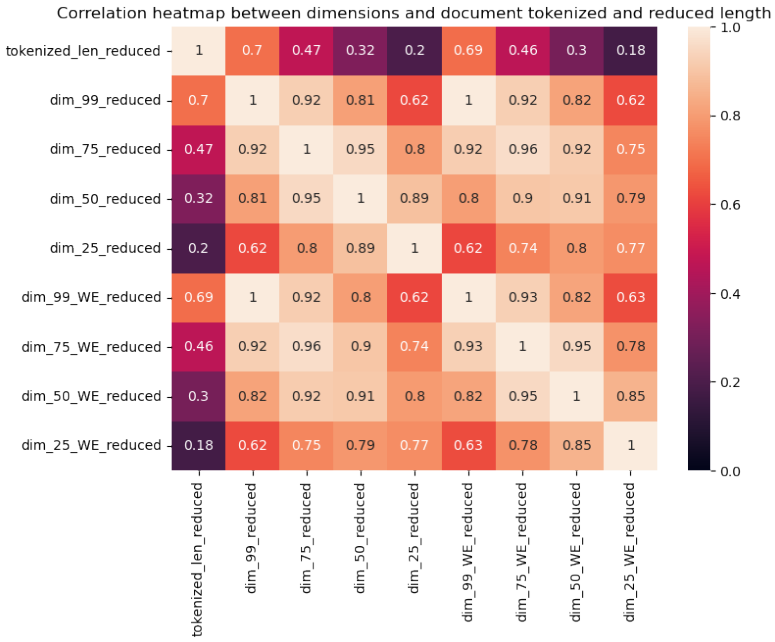


Figura 3.3: Correlación entre el largo del contenido *tokenizado* y reducido de un documento y sus dimensiones cuantificadas a diferentes niveles de varianza.

Se ve que la correlación entre las dimensiones cuantificadas y el tamaño de los documentos disminuye, pero sigue siendo positiva y elevada, especialmente en el caso de varianza al 99 %. Por lo que se probó filtrar los documentos en vez de reducirlos, seleccionando aquellos cuyo largo sea hasta 10 % mayor o menor que la mediana del largo. Al filtrar el *dataset* quedan 284 documentos para el cálculo de dimensiones. En la figura 3.4 se ilustran los cambios.

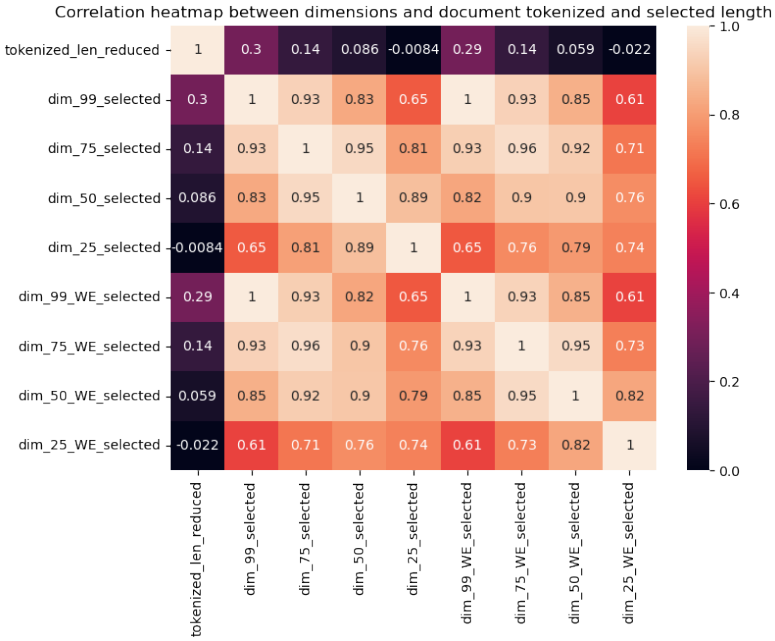


Figura 3.4: Correlación entre el largo del contenido *tokenizado* de documentos similares en tamaño y sus dimensiones cuantificadas a diferentes niveles de varianza.

Los resultados de filtrar que se ven en la figura 3.4 muestran una disminución importante en la correlación, a pesar de que sigue siendo positiva para los casos de 50 % de varianza representada o mayor, pero tiene la desventaja de que reduce el *dataset* considerablemente (de 2225 documentos a 284), por lo que finalmente se decide realizar el mismo análisis de correlación pero mezclando las dos tácticas anteriores, es decir, reduciendo el tamaño de los documentos a la mediana y luego filtrando para seleccionar solamente los documentos que tengan hasta un 10 % de diferencia en tamaño con la mediana del largo. La cantidad de documentos que permanecieron luego del filtro fue de 1970, una cifra mucho más alta que en el filtrado anterior. En la figura 3.5 se visualizan los resultados en un mapa de calor de correlaciones con las dimensiones cuantificadas.

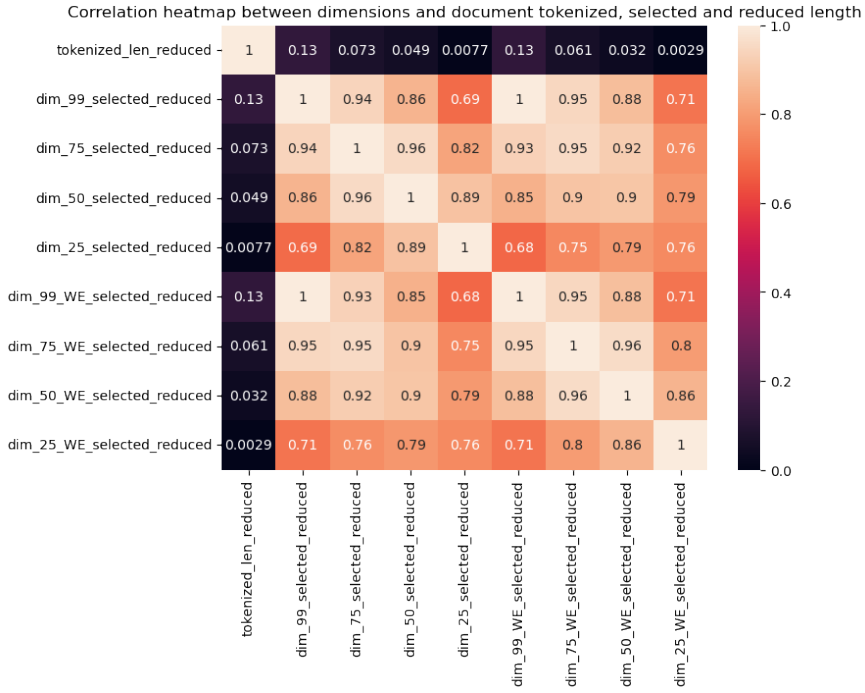
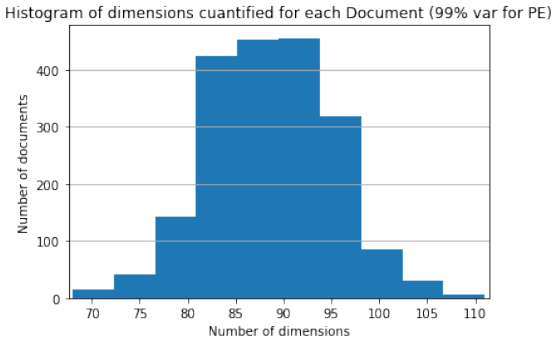


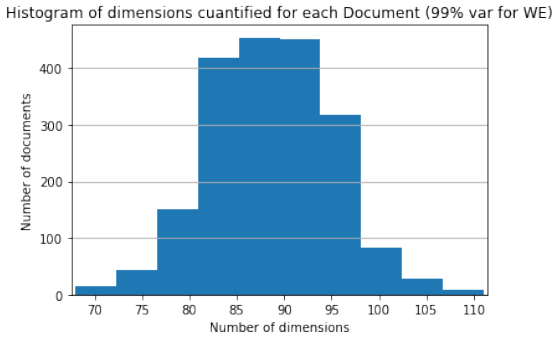
Figura 3.5: En la figura se puede ver la correlación entre el largo del contenido *tokenizado*, reducido y seleccionado y las dimensiones cuantificadas a diferentes niveles de varianza.

Cuando los documentos tienen un largo relativamente similar las correlaciones son bajas, sobre todo para casos de menor varianza representada, como se puede deducir de la figura 3.5. En general, se puede apreciar que las correlaciones son menores cuando se reducen *word embeddings*, pero la diferencia es escasa. El método de reducir y filtrar es el que muestra mejores resultados a pesar de que descarta algunos documentos, éstos son los más pequeños del *dataset* y sólo fueron 55, lo que representa un 11,4 % del conjunto de datos originales.

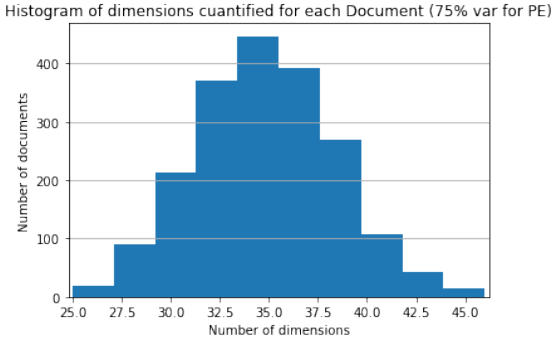
Teniendo decidida la metodología para comparar complejidad entre documentos, se presentan a continuación histogramas para conocer cómo se distribuye la complejidad en el *dataset*. En los gráficos 3.6a, 3.6c, 3.6e y 3.6g se observan los resultados obtenidos al cuantificar las dimensiones reduciendo *embeddings* polares al 99 %, 75 %, 50 % y 25 % respectivamente. Mientras que en los gráficos 3.6b, 3.6d, 3.6f y 3.6h se visualizan las mismas variaciones respectivas, pero reduciendo *word embeddings*.



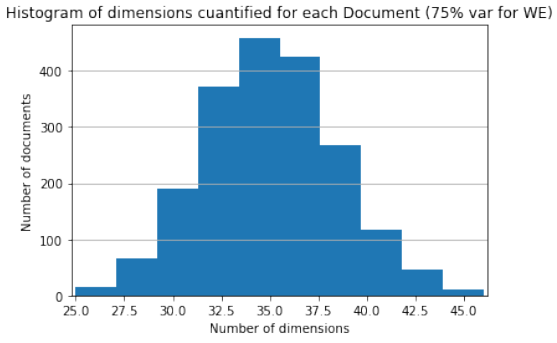
(a) Complejidad al 99 % de var. (PE)



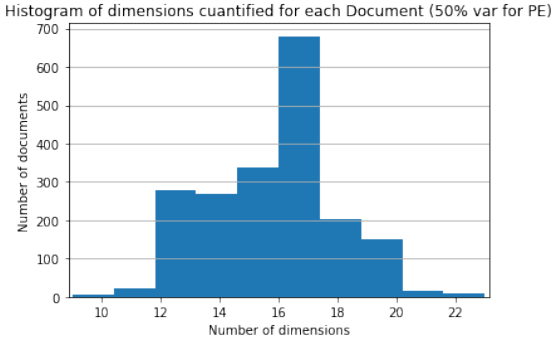
(b) Complejidad al 99 % de var. (WE)



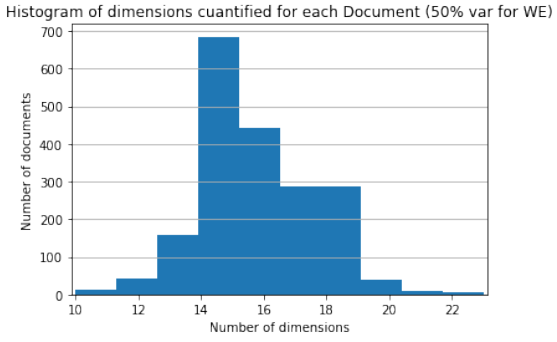
(c) Complejidad al 75 % de var. (PE)



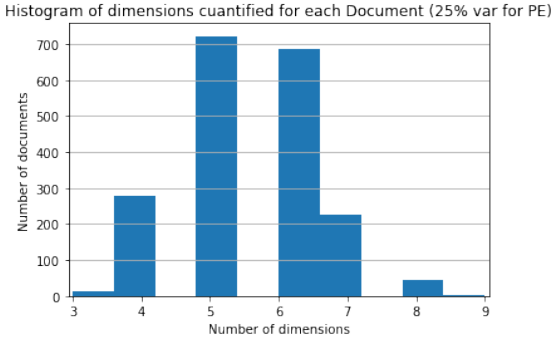
(d) Complejidad al 75 % de var. (WE)



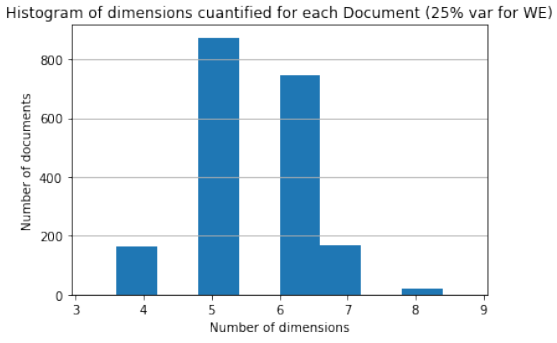
(e) Complejidad al 50 % de var. (PE)



(f) Complejidad al 50 % de var. (WE)



(g) Complejidad al 25 % de var. (PE)



(h) Complejidad al 25 % de var. (WE)

Figura 3.6: Histogramas que ilustran la cantidad de documentos del corpus de noticias de la *BBC* por cada grupo de complejidades, ya sea reduciendo *word embeddings* (WE) o *embeddings* polares (PE) y con distintos porcentajes de variaciones en el análisis de componentes principales.

De la figura 3.6 se puede observar que las distribuciones son muy similares entre *word embeddings* y *embeddings* polares, siendo el caso a 50 % de varianza el más singular por tener las distribuciones más diferentes entre sí. También hay que notar que a mayor varianza es mucho más amplio el rango de complejidades en los que se catalogan los documentos.

3.1.3. Resultados Comparación de Categorías

Siguiendo con el *dataset* modificado con la reducción y filtración que se realizó en la sección anterior, se cuantificó las dimensiones para cada categoría. El procedimiento constó de muestrear 50 documentos aleatorios por cada categoría y cuantificar sus dimensiones con 99 %, 75 % y 50 % de varianza representada. Este cálculo se efectuó 100 veces por cada categoría con la intención de obtener varias mediciones y poder visualizar las distribuciones de los resultados. La cuantificación de dimensiones se realizó reduciendo sobre *embeddings* polares solamente, ya que la intensión es comparar categorías y en la sección anterior se pudo determinar que hay una correlación positiva y alta entre reducir sobre *embeddings* polares y *word embeddings*.

En las figuras 3.7, 3.8 y 3.9 se pueden ver ilustradas las distribuciones de los resultados de dimensiones por cada categoría con gráficos de violín, donde se puede apreciar el máximo, el mínimo, la mediana (marca azul en medio de la línea) y el promedio (punto rojo).

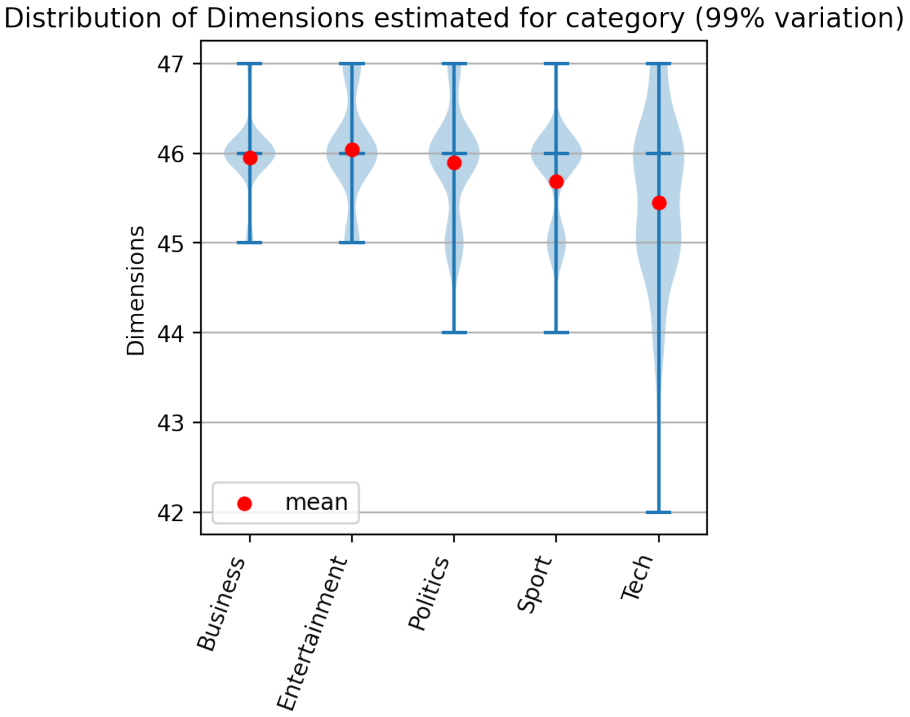


Figura 3.7: Resultado de cuantificar dimensiones por categoría con 99 % de varianza representada.

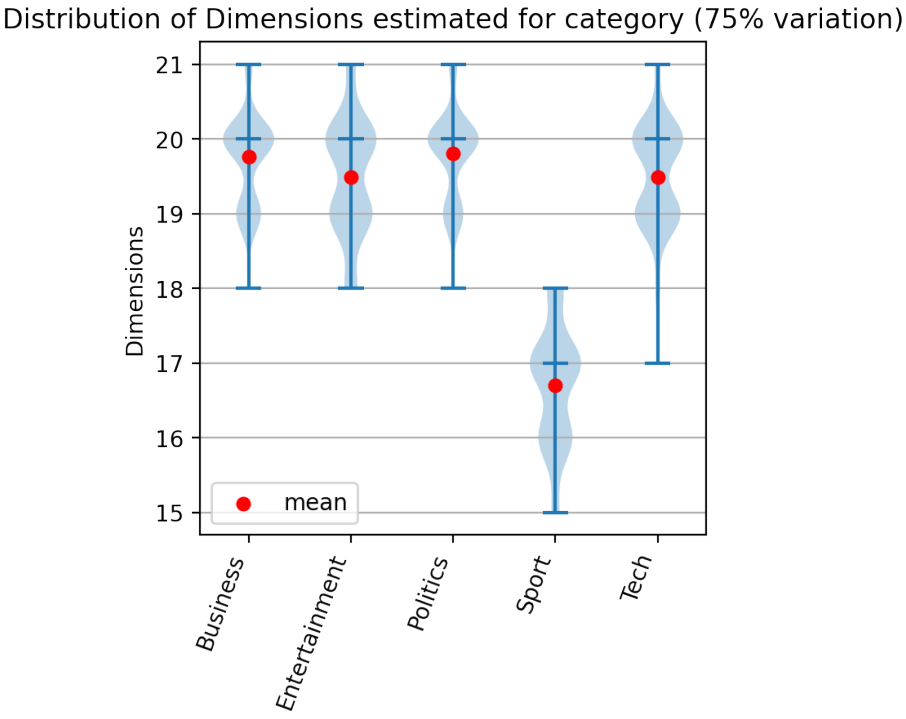


Figura 3.8: Resultado de cuantificar dimensiones por categoría con 75 % de varianza representada.

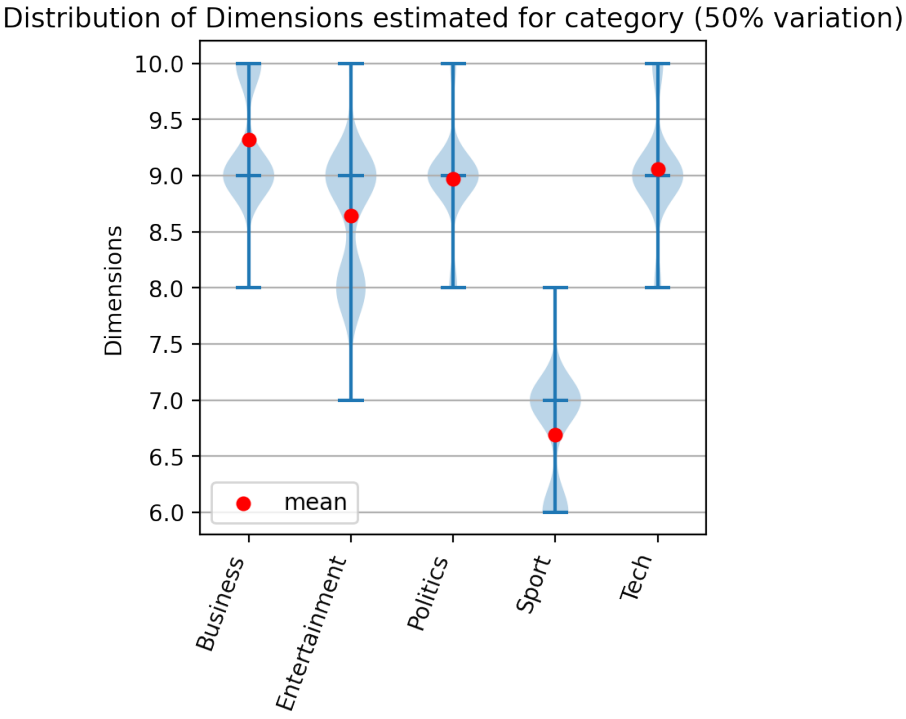


Figura 3.9: Resultado de cuantificar dimensiones por categoría con 50 % de varianza representada.

De los gráficos se puede ver que con una varianza representada del 99 % los resultados son muy similares en términos de máximo, mediana y promedio. Pero viendo las curvas de distribución se puede notar que los resultados de las categorías de Negocios y Entretenimiento están más concentrados cerca de la mediana y el promedio, que son muy parecidos. En cambio, en las categorías de Política, Deportes y Tecnología las distribuciones de los resultados están más esparcidos, sobre todo en tecnología que tiene la mayor diferencia entre mínimo y máximo, y entre promedio y mediana. También se puede ver que a medida que se baja la variación las diferencias se agudizan y son más notorias, tanto así que con 50 % de varianza representada se puede ver claramente como la categoría con menos complejidad es Deportes, mientras que si bien las demás categorías están más cercanas, se resaltan las tendencias ya observadas en el gráfico con 75 % de varianza representada, en que Entretenimiento es la segunda categoría con menor complejidad y Negocios, Tecnología y Política son las más complejas, en ese orden.

3.1.4. Resultados por categoría

Teniendo los resultados de dimensiones de los documentos obtenidas en la sección 3.1.2, por cada categoría se ordenaron los documentos por cantidad de dimensiones estimadas al 99 % de varianza y se segmentaron en dos, de tal manera que dividir cada categoría en documentos de alta complejidad y documentos de baja complejidad. La intuición sobre el concepto de complejidad indica que los documentos de alta complejidad tratan temas más profundos mientras que los de baja complejidad tratan temas más superficiales.

3.1.4.1. Deportes

En las figuras 3.10 y 3.11 se muestran los resultados del *wordcloud* en los documentos de la categoría de Deportes. Entre las diferencias que se pueden apreciar están que en la categoría de alta complejidad tienen más importancia palabras como *win* y en la categoría de baja complejidad aumenta considerablemente la presencia de la palabra *player*. Otras diferencias interesantes de analizar son los crecimientos de las palabras *england* y *would* en los textos de baja complejidad. Una interpretación es que los textos de baja complejidad tienden a hablar más del país de la fuente (Inglaterra) y centrarse en escenarios hipotéticos (aumento de la palabra *would*) mientras que los de alta complejidad se enfocan en informar resultados ocurridos (palabras *win* o *victory* más presentes).

Figura 3.10: *WordCloud* de los documentos de alta complejidad de la categoría Deportes.

Figura 3.11: *WordCloud* de los documentos de baja complejidad de la categoría Deportes.

3.1.4.2. Negocios

En las figuras 3.12 y 3.13 se muestran los resultados del *wordcloud* en los documentos de la categoría de Negocios. Se pueden observar diferencias en palabras que tienen mayor presencia en el *wordcloud* de alta complejidad tales como *economy*, *goverment* o *euro*, mientras que en el de baja complejidad aumentan palabras como *sale*, *profit*, *share*, *new* o *deal*. Estas diferencias pueden interpretarse como que los documentos de la categoría de Negocios de alta complejidad se concentran en temas más cercanos a la macroeconomía y la situación del país, y por otro lado los documentos de baja complejidad se centran en finanzas personales.

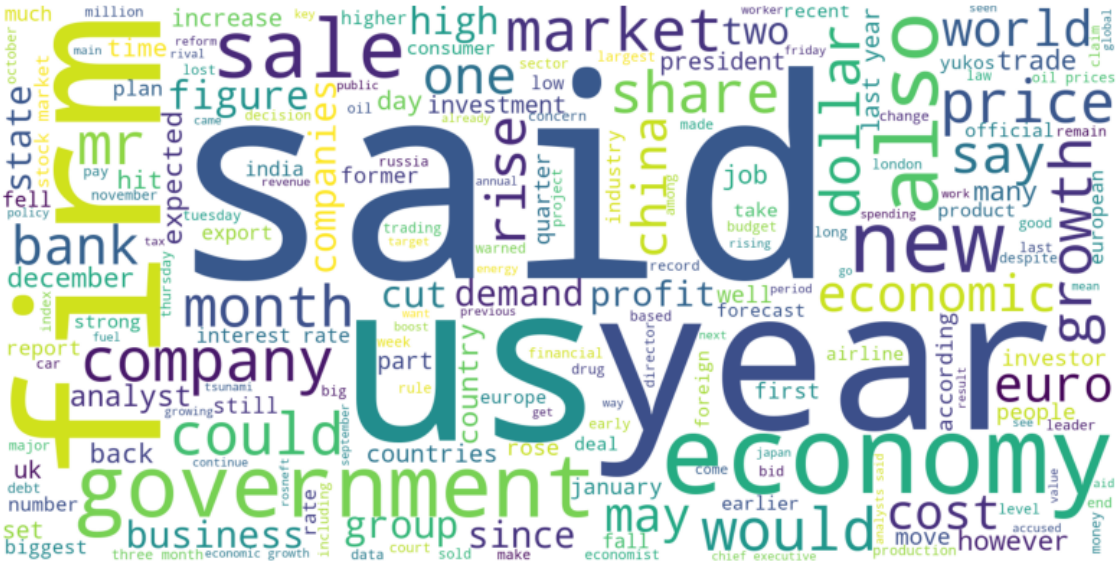


Figura 3.12: *WordCloud* de los documentos de alta complejidad de la categoría Negocios.

Figura 3.13: *WordCloud* de los documentos de baja complejidad de la categoría Negocios.

3.1.4.3. Política

En las figuras 3.14 y 3.15 se muestran los resultados del *wordcloud* en los documentos de la categoría de Política. Se pueden observar diferencias entre los dos *wordclouds* tales como una presencia mayor en el de alta complejidad de palabras como *election*, *britain* o *plan*, mientras que el de baja complejidad palabras como *say*, *mr* o *labour* destacan más que en el de alta complejidad. Lo anterior se puede interpretan como que en los documentos de alta complejidad se tiende a hablar más de procesos políticos a nivel más general, mientras que en los de baja complejidad se hace más énfasis en las declaraciones de individuos del mundo de la política.

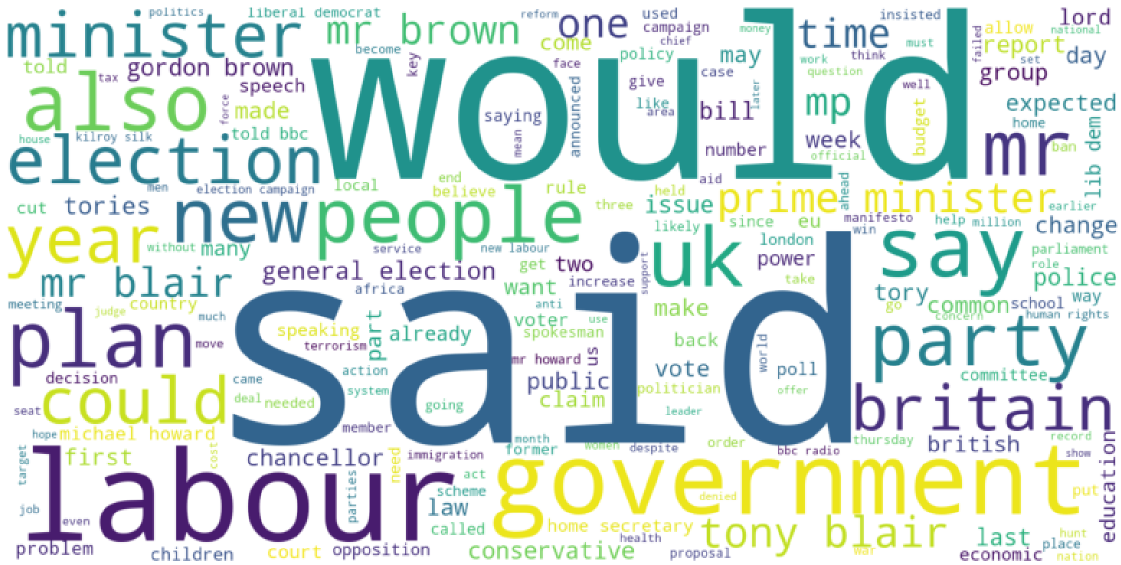


Figura 3.14: *WordCloud* de los documentos de alta complejidad de la categoría Política.

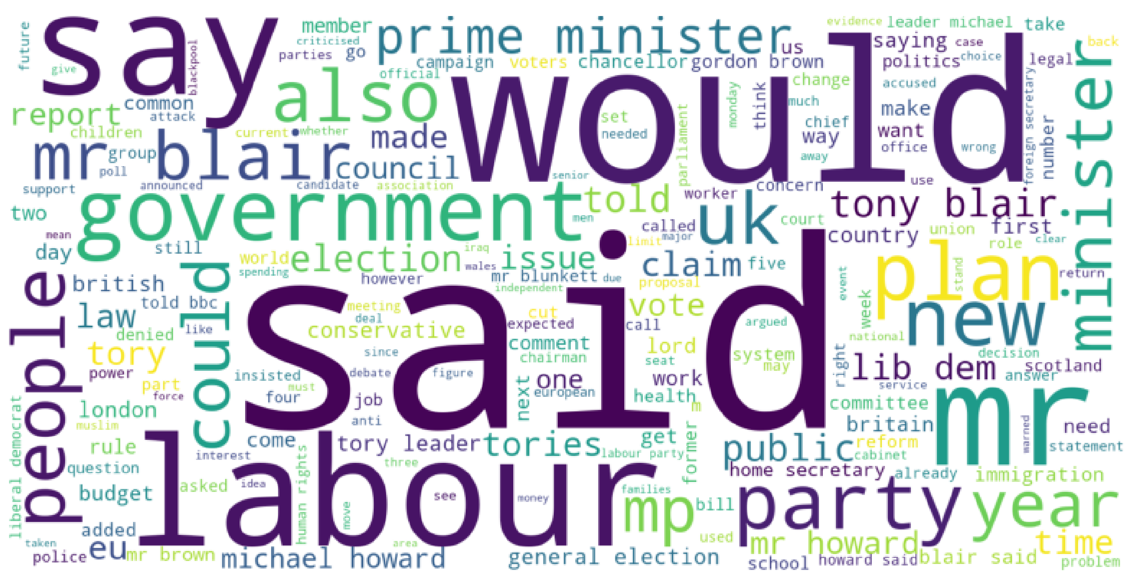


Figura 3.15: *WordCloud* de los documentos de baja complejidad de la categoría Política.

3.1.4.4. Tecnología

En las figuras 3.16 y 3.17 se muestran los resultados de los *wordclouds* en los documentos de la categoría de Tecnología. Se observa que en los documentos de alta complejidad se destacan más palabras como *game*, *software*, *computer* o *technology*. En los documentos de baja complejidad se destacan palabras como *user*, *new*, *mobile* o *firm*. Estas observaciones se pueden interpretar como que en los documentos más complejos de esta categoría se tiende a hablar más de tecnologías concretas (uso de palabras como *software* o *computer*), y en los de baja complejidad se tiende a describir de donde provienen las nuevas tecnologías y quiénes las usan.

Figura 3.16: *WordCloud* de los documentos de alta complejidad de la categoría Tecnología.

Figura 3.17: *WordCloud* de los documentos de baja complejidad de la categoría Tecnología.

3.1.4.5. Entretenimiento

En las figuras 3.18 y 3.19 se muestran los resultados del *wordcloud* en los documentos de la categoría de Entretenimiento. Se aprecia que en los documentos de alta complejidad sobresalen palabras como *star*, *hit* o *director*. Mientras que en los de baja complejidad tienen más frecuencia palabras como *award*, *new* o *music*. Se puede interpretar que en los documentos de baja complejidad se habla más de premiaciones o novedades y en los de alta complejidad pareciera que se describen temas relacionados a celebridades o artistas famosos. A diferencia de las categorías anteriores, en ésta no parece haber una diferencia temática notoria.

Figura 3.18: *WordCloud* de los documentos de alta complejidad de la categoría Entretenimiento.

Figura 3.19: *WordCloud* de los documentos de baja complejidad de la categoría Entretenimiento.

3.1.5. Relación de los componentes principales con *Bias* e *Intensity*

Como los componentes principales son combinaciones lineales de las dimensiones originales, al efectuar un análisis de componentes principales sobre *embeddings* polares se puede obtener lo que se conoce en inglés como *loading scores*, que son los pesos de cada una de las dimensiones originales que conforman las nuevas dimensiones. Con estos pesos se puede interpretar cuáles son las dimensiones más importantes para producir cada componente principal.

En esta sección se comparará los *loading scores* con las mediciones de *Bias* e *Intensity* propuestas por *FrameAxis*. Para esto, primero se calculó los valores de *Bias* e *Intensity* para todos los ejes semánticos. Después se consideraron los tres primeros componentes principales, pues al ser los que representan mayor cantidad de varianza son los más relevantes en el cambio de eje de coordenadas. Y se consideraron la suma de los *loading scores* y la suma ponderada de los mismos. La idea es tener una variable que represente de manera general los pesos de todos los componentes principales. Teniendo como variables *Bias*, *Intensity*, los tres primeros componentes principales, la suma y la suma ponderada de los *loading scores* para cada eje semántico se produce el siguiente mapa de calor de la correlación entre las variables mencionadas en la figura 3.20.

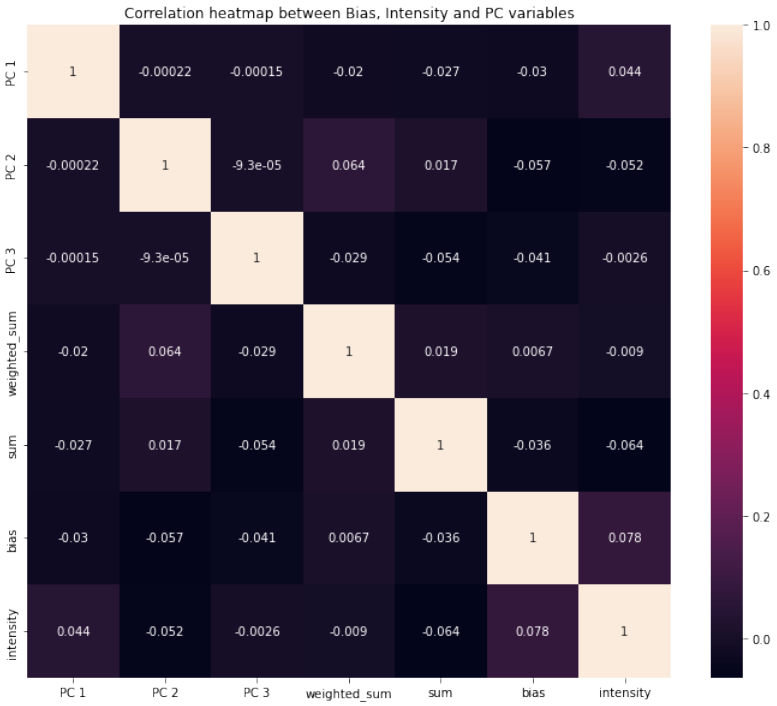


Figura 3.20: Mapa de calor de la correlación entre *Bias*, *Intensity*, los tres primeros componentes principales, la suma y la suma ponderada de los *loading scores*.

De la figura 3.20 se puede ver que los valores de correlación son tan cercanos al 0 en todas las comparaciones que se puede interpretar que no hay correlación significativa entre ninguna de las variables que componen el análisis.

3.1.6. Comparación de la varianza aportada por los componentes principales por categoría

Gracias al análisis de componentes principales se puede saber cuánto porcentaje de varianza aporta cada componente principal al cambio de eje de coordenadas. Al graficar estos porcentajes para cada categoría se puede comparar qué categorías tienen los primeros componentes que aportan mayor varianza a la representación de ésta y cómo van decreciendo en los que preceden.

Para realizar esta comparativa se muestrearon 50 documentos aleatorios por cada categoría del conjunto de datos y se cuantificaron sus dimensiones con varianza representada de 99 % ya que en este caso permite ilustrar todos los datos.

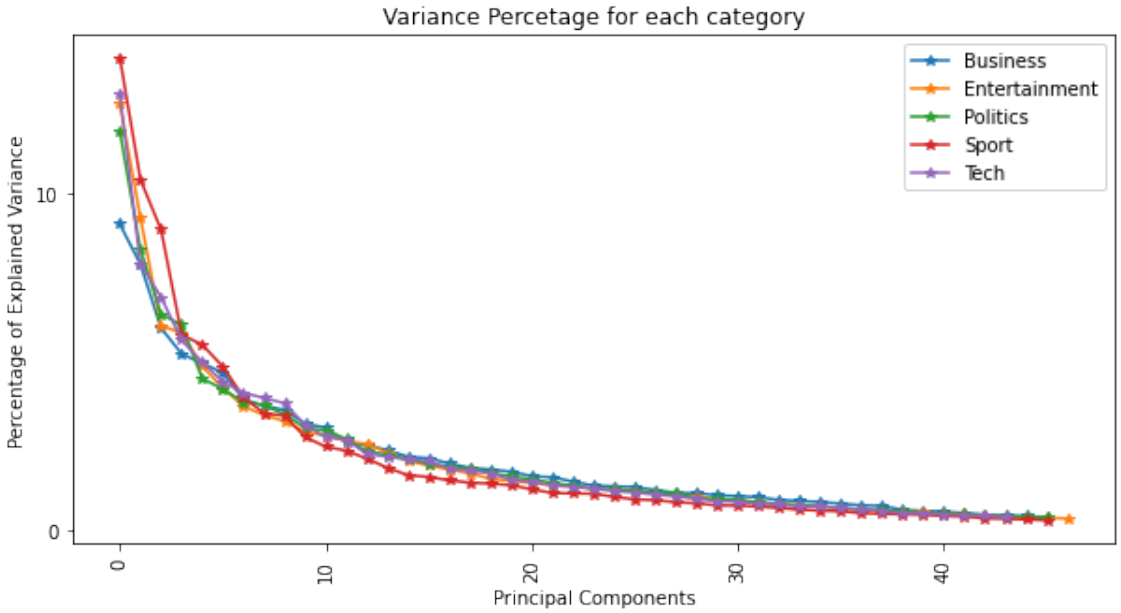


Figura 3.21: Gráfico de porcentaje de varianza aportada por cada componente principal en cada categoría.

La figura 3.21 expone cómo se comporta el porcentaje de varianza para cada categoría, se nota que la categoría de Deportes es la que tienen el primer componente principal con más varianza representada, mientras que la categoría de Negocios es la que tiene menos varianza representada con el primer componente. Por la forma que tiene el decrecimiento de las curvas trazadas se realizó un gráfico que suma las varianzas acumuladas para poder ver cómo crece la representación de varianza con escala logarítmica en el eje x e y. Esto se puede visualizar en la figura 3.22.

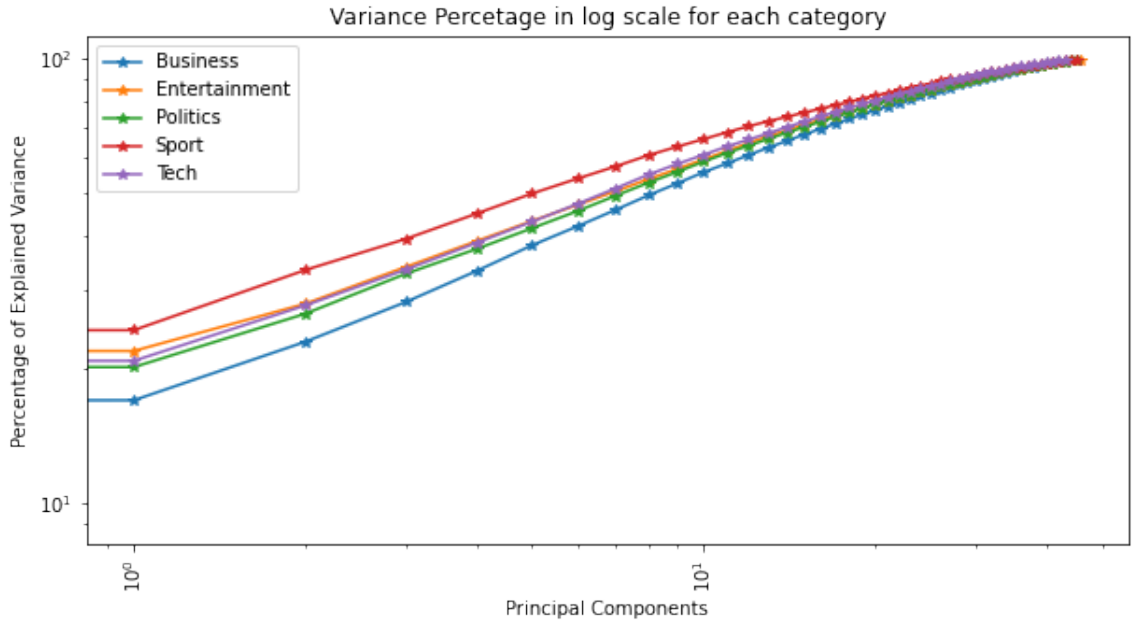


Figura 3.22: Gráfico de porcentaje de varianza acumulada por los componentes principales en cada categoría en escala logarítmica.

En la figura 3.22 se puede observar cómo crece la representación de varianza a medida que se avanza por los componentes principales. Es importante destacar que las categorías de Deportes y Entretenimiento son las que más varianza representan en sus primeros componentes, siendo justamente las categorías que tienen menores resultados de complejidad como se vio en la sección 3.1.3. Así mismo, como se vio que las categorías de Negocios y Política son las más complejas, en este gráfico son las que tienen menor varianza representada en sus primeros componentes principales. Para este gráfico analizar los primeros componentes principales es lo más significativo porque todos las curvas se dirigen al 100 % de varianza representada.

Capítulo 4

Evaluación

En este capítulo se realizan dos tipos de evaluaciones para comprobar la utilidad del método desarrollado. En la sección 4.1 se realiza una evaluación cualitativa para entender las diferencias entre un texto que el método determina que es de alta complejidad comparado con un texto etiquetado como de baja complejidad. Y en la sección 4.2 se efectúan dos tareas típicas en procesamiento de lenguaje natural para evaluar la *performance* de los resultados a través de esas tareas.

4.1. Evaluación Cualitativa

Aprovechando la partición de los documentos del *dataset* de noticias de la *BBC* en documentos de complejidad alta y baja utilizado en la sección 3.1.4 se realizó una evaluación cualitativa para poder ejemplificar los resultados de complejidad en los documentos. Para esto se escogieron el texto más complejo y el menos complejo del *dataset*, y así poder comparar diferencias cualitativas en su lectura. Dada la diferencia de tamaño en los textos que se presentan a continuación (que se presentan íntegros), cabe destacar que el tamaño de los textos no influyó en la cuantificación de su complejidad pues los textos más largos fueron acortados y los más cortos fueron filtrados como se determinó en la sección 3.1.2.

4.1.1. Documento de alta Complejidad

El documento más complejo se titula *The Force is strong in Battlefront*, es de la categoría Tecnología y tiene una complejidad de 111 dimensiones. Su contenido es el siguiente:

> *The warm reception that has greeted Star Wars: Battlefront is a reflection not of any ingenious innovation in its gameplay, but of its back-to-basics approach and immense nostalgia quotient. Geared towards online gamers, it is based around little more than a series of all-out gunfights, set in an array of locations all featured in, or hinted at during, the two blockbusting film trilogies. Previous Star Wars titles like the acclaimed Knights Of The Old Republic and Jedi Knight have regularly impressed with their imaginative forays into the far corners of the franchise's extensive universe, and their use of weird and wonderful new characters.*

Battlefront on the other hand wholeheartedly revisits the most recognisable elements of the hit movies themselves. The sights, sounds and protagonists on show here will all be instantly familiar to fans, who may well feel that the opportunity to relive Star Wars' most memorable screen skirmishes makes this the game they have always waited for. The mayhem can be viewed from either a third or first-person perspective, and you can either fight for the forces of freedom or join Darth Vader on the Dark Side, depending on the episode and type of campaign as well as the player's personal propensity for good or evil. There is ample chance to be a Wookie, shoot Ewoks and rush into battle alongside a fired-up Luke Skywalker. In each section, the task is simply to wipe out enemy troops, seize strategic waypoints and move on to the next planet. It really is no more complicated than that. Locations include the frozen wastes of Hoth, the ice planet from The Empire Strikes Back, complete with massive mechanical AT-ATs on the march. There are also the dusty, sinister deserts of Tatooine and Geonosis, as well as the forest moon of Endor, where Return Of The Jedi's much-maligned Ewoks lived. The feel of those places is well and truly captured, with both backdrops and characters looking good and very authentic. It is worth noting though that on the PlayStation 2, the game's graphics are a curiously long way behind those of the Xbox version. The pivotal element behind Battlefront's success is that it successfully gives you the feel of being of being plunged into the midst of large-scale war. The number of combatants, noise and abundance of laser fire see to that, and the sense of chaos really comes over. Speaking of noise, Battlefront is a real testament to the strength of the Star Wars galaxy's audio motifs. The multitude of distinctive weapon and vehicle noises are immensely familiar, as are the stirring John Williams symphonies that never let up. There is also a particularly snazzy remix of one of his themes in the menu section. It has to be said if the game did not have the boon of being Star Wars, it would not stand up for long. The gameplay is reliable, bog-standard stuff, short on originality. There are also odd annoyances, like the game's insistence on re-spawning you miles away from the action, an irritating price to pay for not getting blown up the second you appear. And some of the weapons and vehicles are not as responsive and fluid to operate as they might be. That said, it is still great fun to pilot a Scout Walker or Speeder Bike, however non user-friendly they prove. Whilst it is firmly designed with multiplayer action in mind, Battlefront is actually perfectly good fun as an offline game. The above-average AI of the enemy sees to that, although given the frenetic environments they operate in, their strategic behaviour does not need to be all that sophisticated. Battlefront's novelty value will doubtless wear off relatively fast, leaving behind a slightly empty one-trick-pony of a game. But for a while, it is an absolute blast, and one of the most immediately satisfying video game offerings yet from George Lucas' stable.

A continuación, se presentan tres tablas que muestran los pares polares más importantes en la composición de los componentes principales del documento de alta complejidad. La tabla 4.1 muestra los pares polares que tienen los pesos más altos en la composición del primer componente principal (el que aporta mayor varianza), la tabla 4.2 muestra los pares polares que tienen mayor participación sumando todos los pesos de todos los componentes principales y la tabla 4.3 tiene los pares polares más relevantes al sumar los pesos que tienen

en componente principal ponderado por la varianza que aportan los componentes.

Pares polares	**PC 1**
meat, plant	-0.048603
blow, breathe	0.048067
copier, innovator	-0.047935
crab, gregarious	0.047768
polish, roughen	-0.047644
irritate, please	0.047632
harvest, plant	-0.047417
hail, snow	0.046865
astronomic, geologic	0.046447
color, dull	-0.046023

Tabla 4.1: Primeros 10 pares polares en documento de alta complejidad.

Pares polares	**Suma**
implicate, save	0.742206
steel, weak	0.734290
befuddle, deduce	0.729900
flute, trombone	0.727917
shirk, toil	0.727873
favorable, unfavorable	0.723655
confirmation, contradiction	0.714956
modal, unnatural	0.711385
age, young	0.709526
bind, loose	0.704798

Tabla 4.2: Primeros 10 pares polares en documento de alta complejidad según suma de los componentes principales.

Pares polares	**Suma ponderada**
enliven, immobilize	0.024808
satisfactory, unsatisfactory	0.024433
blandness, sensory	0.024181
both, one	0.024170
fragile, strong	0.024134
musician, mute	0.024114
harsh, symphonic	0.0241007
straightforward, symbolic	0.024060
change, stationary	0.024054
get, snowball	0.024044

Tabla 4.3: Primeros 10 pares polares en documento de alta complejidad según suma de los componentes principales ponderados en la varianza que aportan

4.1.2. Documento de baja Complejidad

El documento menos complejo se titula *Irish finish with home game*, es de la categoría Deportes y tiene una complejidad de 68 dimensiones. Su contenido es el siguiente:

> *Republic of Ireland manager Brian Kerr has been granted his wish for a home game as the final World Cup qualifier. Ireland will close their bid to reach the 2006 finals by playing Switzerland in Dublin on 12 October 2005. The Republic met the Swiss in their final Euro 2004 qualifier, losing 2-0 away and missing out on a place in the finals in Portugal. The Group Four fixtures were hammered out at a meeting in Dublin on Tuesday. The Irish open their campaign on 4 September at home to Cyprus and wrap up the 10-match series on 12 October 2005, with the visit of Switzerland. Manager Brian Kerr and FAI officials met representatives from Switzerland, France, Cyprus, Israel and the Faroe Islands to arrange the fixture schedule. Kerr had hoped to finish with a clash against France, but got the reigning European champions as their penultimate home match on 7 September 2005. The manager got his wish to avoid a repeat of finishing their bid to qualify with too many away matches. Republic of Ireland v Cyprus; France v Israel; Switzerland v Faroe Islands. Switzerland v Republic of Ireland; Israel v Cyprus; Faroe Islands v France. France v Republic of Ireland; Israel v Switzerland; Cyprus v Faroe Islands. Republic of Ireland v Faroe Islands; Cyprus v France. Cyprus v Israel. France v Switzerland; Israel v Republic of Ireland. Switzerland v Cyprus; Israel v France. Republic of Ireland v Israel; Faroe Islands v Switzerland. Faroe Islands v Republic of Ireland. August 17 - Faroe Islands v Cyprus. France v Faroe Islands; Switzerland v Israel. Republic of Ireland v France; Cyprus v Switzerland; Faroe Islands v Israel. Switzerland v France; Israel v Faroe Islands; Cyprus v Republic of Ireland. France v Cyprus; Republic of Ireland v Switzerland.*

Tal como en la noticia anterior, se presentan las tablas 4.4, 4.5 y 4.6 que muestran los pares polares más importantes para el primer componente principal, para la suma de los pesos que componen los componentes principales y para la suma de los pesos ponderada en la varianza que aporta cada componente principal.

Pares polares	**PC 1**
meat, plant	-0.050651
harvest, plant	-0.050145
polish, roughen	-0.049253
hail, snow	0.049065
crab, gregarious	0.048916
copier, innovator	-0.048600
plant, rock	0.048244
metal, paper	-0.048155
color, dull	-0.048069
blow, breathe	0.047941

Tabla 4.4: Primeros 10 pares polares en documento de baja complejidad.

Pares polares	**Suma**
call, text	0.748685
chair, stretcher	0.691640
fool, smart	0.685608
bind, loose	0.684661
learn, stupid	0.664176
expose, hedge	0.661711
sheep, wolf	0.652208
defamation, promotional	0.652134
unemployment, workforce	0.650745
muscular, neural	0.648312

Tabla 4.5: Primeros 10 pares polares en documento de baja complejidad según suma de los componentes principales ponderados en la varianza que aportan

Pares polares	**Suma ponderada**
cavity, fill	0.025049
solo, symphonic	0.024624
critic, musician	0.024388
friendship, rivalry	0.024227
alphabet, count	0.024190
accountant, musician	0.024088
unemployed, workforce	0.024081
exchange, keep	0.024076
oceanic, shallow	0.024049
low, tall	0.024043

Tabla 4.6: Primeros 10 pares polares en documento de baja complejidad según suma de los componentes principales.

Al comparar el contenido de ambos documentos es notoria la diferencia en complejidad que se captura, mientras que la noticia de alta complejidad toca tópicos como los videojuegos, las películas y la música, ofreciendo opiniones diferentes para cada tema; el documento de baja complejidad contiene información objetiva y simplemente lista una serie de eventos. Desde el punto de vista del lenguaje utilizado el documento de alta complejidad utiliza palabras más específicas y cultas para elaborar su relato, junto con palabras que corresponden a distintas áreas, como términos relacionados a la guerra, el universo, las películas, los videojuegos o la música; en cambio el documento de baja complejidad utiliza en su mayoría palabras relacionadas con el deporte y los países. Estos ejemplos reflejan resultados que concuerdan con la evaluación humana de complejidad, en tanto se usen las dimensiones para comparar distintos textos. Se presentan más ejemplos de noticias de alta y baja complejidad en el anexo de este informe.

Con respecto a las tablas presentadas, primero mencionar que en las tablas 4.1 y 4.4 se muestran los pares polares (o ejes semánticos) que más peso tienen ya sea positivo o negativo, y que si es positivo significa una inclinación hacía la palabra de la derecha, mientras que negativo se inclina hacía la palabra de la izquierda. En cambio, para en las tablas 4.2, 4.3,

4.5 y 4.6 sólo se presentan valores positivos pues su objetivo es tratar de describir los ejes semánticos que más participación y aporte tienen para crear los componentes principales, por lo que se construyen sumando valores absolutos de los pesos, dado esto no se puede interpretar una inclinación hacia una de las palabras.

Al analizar los resultados de las tablas del documento de alta complejidad, se ve que en la tabla 4.1 aparecen ejes semánticos difíciles de relacionar con el texto como *meat-plant*, mientras que también hay algunos que sí se pueden relacionar con algo hablado en el texto, como *irritate-please*, *polish-roughen* o *astronomic-geologic*. Así mismo se pueden encontrar en la tabla 4.2 ejes semánticos que hacen sentido con los tópicos y opiniones del texto como *steel-weak*, *confirmation-contradiction* o *favorable-unfavorable*. Y lo mismo para la tabla 4.3, donde se encuentran *satisfactory-unsatisfactory*, *musician-mute* o *straightforward-symbolic*. El mismo análisis para el documento de baja complejidad da cuenta que es mucho más difícil encontrar ejes semánticos que hagan sentido con el texto en cualquiera de las tablas, más allá de los pares *unemployed-workforce* y *friendship-rivalry* de la tabla 4.6. Todas estas observaciones indican que los ejes semánticos obtenidos no son concluyentes ni ofrecen gran ayuda, pero no sólo por los resultados obtenidos sino también por la composición de los pares polares, que al igual que en el *framework POLAR* son obtenidos de una fuente especializada llamada *WordNet*.

4.2. Evaluación Cuantitativa

En esta sección se describen dos tares típicas del procesamiento del lenguaje natural: Clasificación de Noticias y Análisis de Sentimiento. El objetivo es comparar el rendimiento de representaciones vectoriales de las palabras utilizando *word embeddings* y *embeddings* polares, tanto íntegros (con 300 dimensiones en el caso de los *word embeddings*) como reducidos con análisis de componentes principales para distintos niveles de varianza representada. Los *embeddings* base para este análisis fueron *word embeddings* preentrenados de *GloVe* de 300 dimensiones con 6 billones de palabras. Mientras que se usaron los mismos pares polares utilizados en la sección 3 para construir los *embeddings* polares, pero en este caso fueron reducidos a 500 dimensiones polares, a través del método de selección por maximización de la varianza (descrito en 1.3.3), para entrenar los modelos más rápidamente. Los resultados que se muestran en las siguientes secciones corresponden al mejor puntaje de *accuracy* obtenido por cualquiera de los modelos que se describirán dado un conjunto de *features*, y todos los resultados son comparados con los *embeddings* sin reducir, por lo que estos últimos se etiquetan como 100 % de variación en las visualizaciones. Para entrenar los modelos se usaron *features* generados a partir de *word embeddings* y *embeddings* polares en representaciones de varianza de 100 % (sin reducción dimensional), 99 %, 90 %, 80 %, 75 %, 70 %, 60 %, 50 %, 40 %, 30 %, 25 %, 20 % y 10 %

4.2.1. Evaluación en tarea de clasificación

En este caso, de la misma manera que en *POLAR*, se realizaron tres tareas de clasificaciones binarias con el *dataset 20 Newsgroups* [1] [7] que se puede obtener a través de la librería *sklearn*. En las tres tareas se compartió la metodología que consistió en utilizar como *features*

[1] `http://qwone.com/~jason/20Newsgroups/`

representaciones vectoriales de las noticias para entrenar distintos modelos de clasificación. Estas representaciones vectoriales fueron los promedios de los vectores que representaban las palabras. Los modelos de clasificación utilizados fueron *Support Vector Classification*, *Gaussian Naive Bayes*, *Multi-layer Perceptron classifier* y *Random Forest Classifier*, todos de la librería *sklearn*.

4.2.1.1. Clasificación Tecnología

La primera clasificación tiene como objetivo determinar si una noticia de computadores involucra *IMB* o *Mac*. Se dividieron los datos entre 929 noticias para entrenamiento, 239 para validación y 777 para testeo. Luego de entrenar los clasificadores para *word embeddings* y *embeddings* polares en distintas representaciones de varianza se obtuvieron los resultados graficados en la figura 4.1.

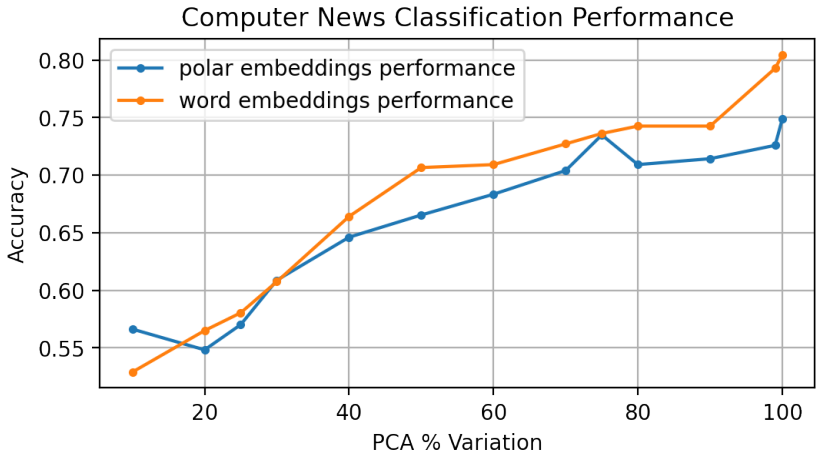


Figura 4.1: Gráfico de *performance* de los *embeddings* reducidos en tarea de clasificación de noticias de computadores.

Se puede comprobar que en general los *embeddings* polares tienen un rendimiento menor que los *word embeddings*, pero muy pequeña en algunos tramos. Para varianzas representadas menores a 70 % se obtienen puntajes de *accuracy* menores al 70 % para *embeddings* polares y lo mismo para *word embeddings* desde el 60 % de la varianza representada. Es importante notar que el rendimiento disminuye muy poco para 99 % de la varianza representada porque ese pequeño cambio en varianza representada es un salto grande en la cantidad de dimensiones disminuidas, por lo que en las figuras 4.2 y 4.3 se visualiza la *performance* en comparación con la cantidad de dimensiones de los *embeddings*.

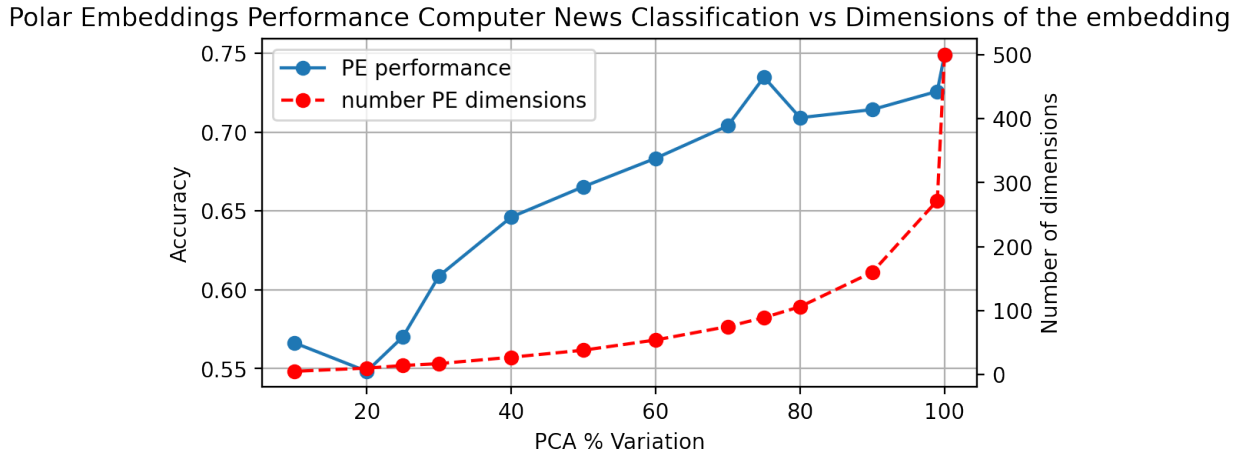


Figura 4.2: Gráfico de *performance* de los *embeddings* polares reducidos en tarea de clasificación de noticias de computadores versus las dimensiones de los *embeddings* polares.

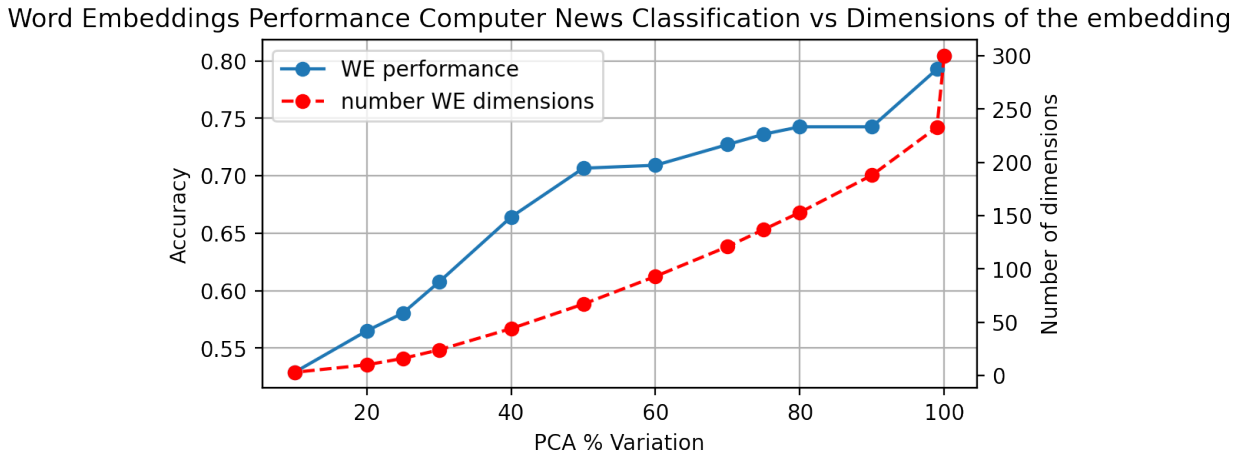


Figura 4.3: Gráfico de *performance* de los *word embeddings* reducidos en tarea de clasificación de noticias de computadores versus las dimensiones de los *word embeddings*.

En las figuras 4.2 y 4.3 se aprecia que las dimensiones de los *embeddings* decrecen rápidamente a medida que decrece la varianza representada y que sus mayores saltos se encuentran en las primeras disminuciones de varianza representada, significando que una pequeña reducción de los *embeddings* mantiene una *performance* similar a los *embeddings* completos.

4.2.1.2. Clasificación Religión

La segunda clasificación consta de predecir si una noticia religiosa involucra ateísmo o cristianismo. Se dividieron los datos entre 870 noticias para entrenamiento, 209 para validación y 717 para testeo. Luego de entrenar los clasificadores para *word embeddings* y *embeddings* polares en diferentes representaciones de varianza se obtuvieron los resultados graficados en la figura 4.4. Además, en las figuras 4.5 y 4.6 se ilustra la *performance* en contraste con la cantidad de dimensiones de los *embeddings*.

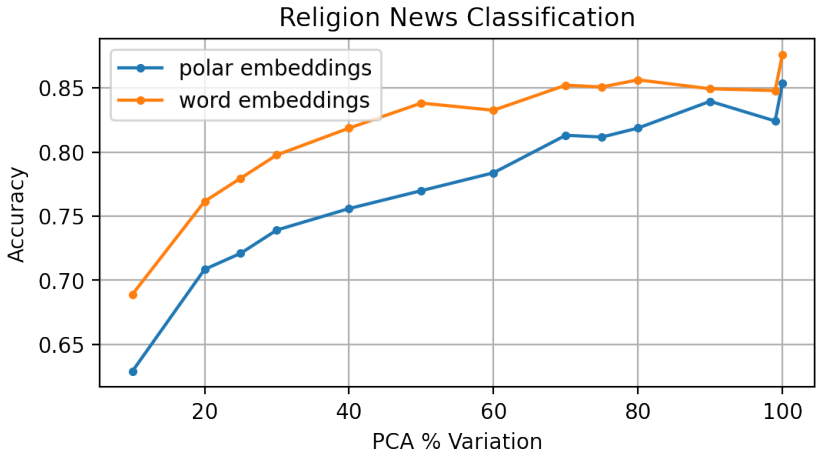


Figura 4.4: Gráfico de *performance* de los *embeddings* reducidos en tarea de clasificación de noticias de religión.

En la figura 4.4 se aprecia un compartamiento similar a la clasificación anterior, donde los *word embeddings* tienen mejores resultados, pero por una diferencia pequeña, pero se diferencia en que los *embeddings* polares se mantienen más cerca en las varianzas representadas más altas. Y también en que las curvas decrecen menos pronunciadamente.

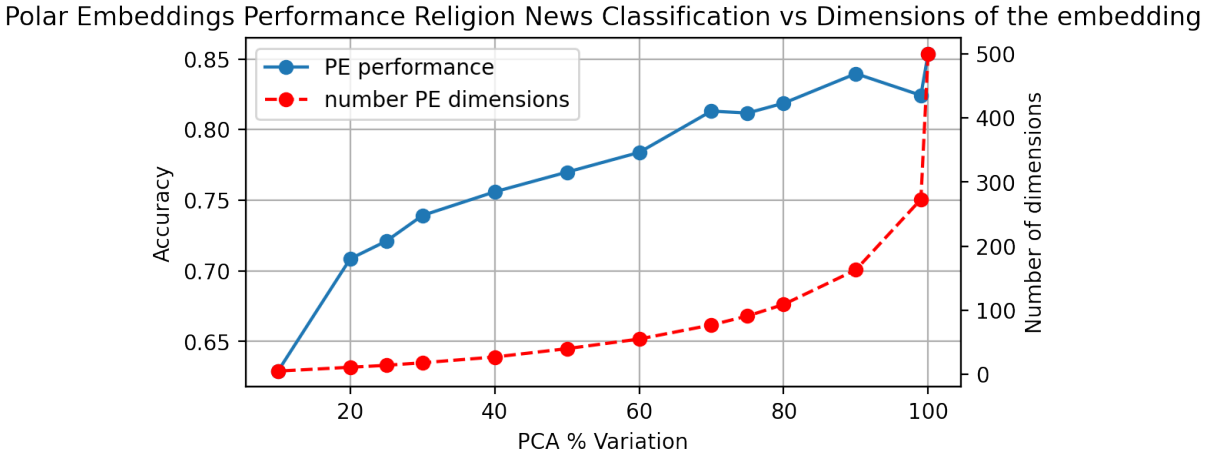


Figura 4.5: Gráfico de *performance* de los *embeddings* polares reducidos en tarea de clasificación de noticias de religión versus las dimensiones de los *embeddings* polares.

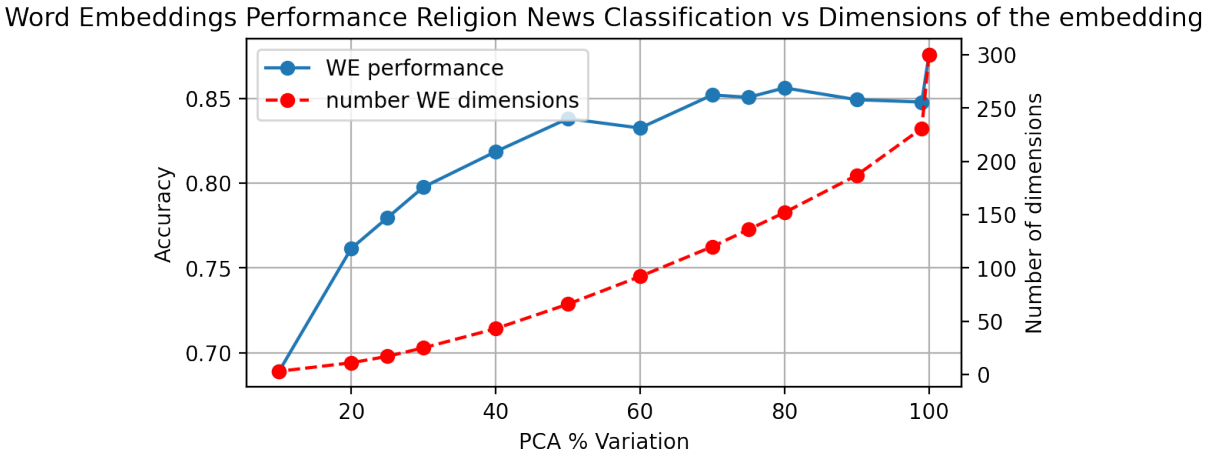


Figura 4.6: Gráfico de *performance* de los *word embeddings* reducidos en tarea de clasificación de noticias de religión versus las dimensiones de los *word embeddings*.

Además, en las figuras 4.5 y 4.6 se pueden ver nuevamente comportamientos similares a la clasificación anterior ya que las dimensiones disminuyen rápidamente y más notoriamente en las primeras bajas en la varianza representada.

4.2.1.3. Clasificación Deportes

La primera clasificación consiste en categorizar entre hockey o béisbol un conjunto de noticias de deportes. Para esta tarea se dividieron los datos entre 958 noticias para entrenamiento, 239 para validación y 796 para testeo. Luego de entrenar los clasificadores para *word embeddings* y *embeddings* polares en diferentes representaciones de varianza se obtuvieron los resultados graficados en la figura 4.7. Además, en las figuras 4.8 y 4.9 se visualiza la *performance* en contraste con la cantidad de dimensiones de los *embeddings*.

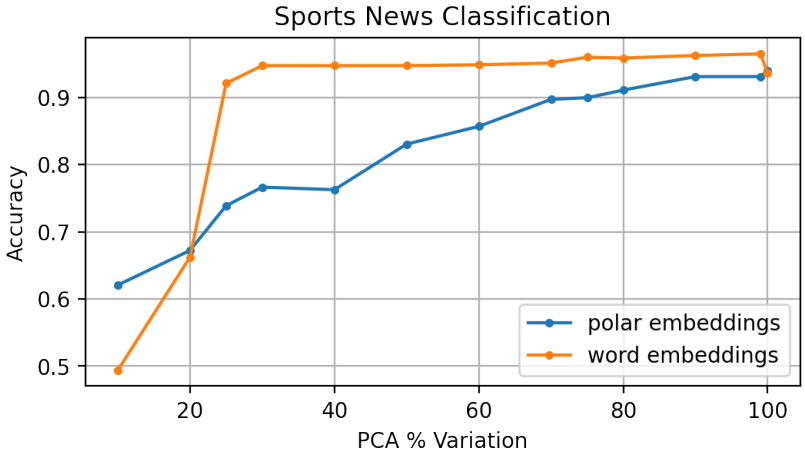


Figura 4.7: Gráfico de *performance* de los *embeddings* reducidos en tarea de clasificación de noticias de deportes.

La figura 4.7 muestra un comportamiento diferente a las clasificaciones anteriores en la

performance de los *word embeddings* porque se mantienen con un *accuracy* mayor al 90 % hasta con 25 % de varianza representada. Mientras que los *embeddings* polares sí disminuyen de una manera similar a las vistas en los dos casos anteriores.

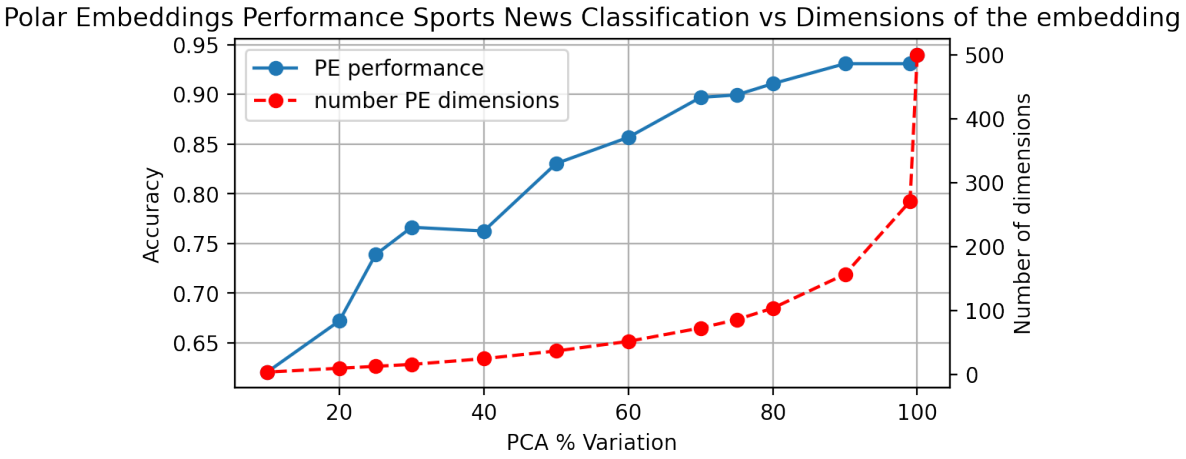


Figura 4.8: Gráfico de *performance* de los *embeddings* polares reducidos en tarea de clasificación de noticias de deportes versus las dimensiones de los *embeddings* polares.

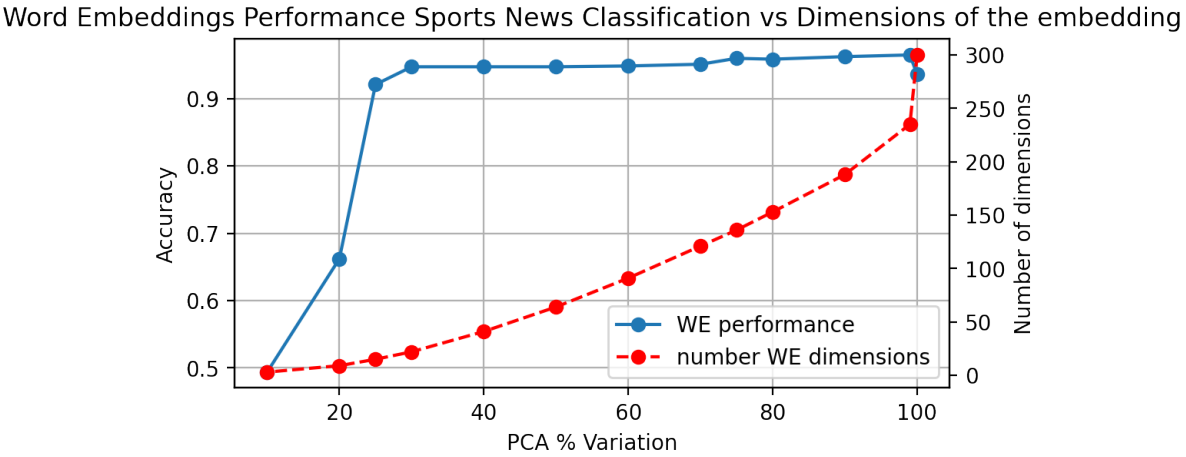


Figura 4.9: Gráfico de *performance* de los *word embeddings* reducidos en tarea de clasificación de noticias de deportes versus las dimensiones de los *word embeddings*.

Se puede ver en las figuras 4.8 y 4.9 que las dimensiones disminuyen tal como en los casos de clasificación anteriores, decreciendo rápidamente y con sus mayores saltos en las primeras bajas en la varianza representada.

4.2.2. Evaluación en tarea de análisis de sentimientos

Del mismo modo que en *POLAR* se realizó una tarea de análisis de sentimiento con el *dataset Stanford Sentiment Treebank* [16] con el objetivo de clasificar una oración dada como positiva o negativa. El conjunto de datos se dividió en 6920 oraciones para entrenamiento,

872 para validación y 1821 para testeo. La metodología utilizada consistió en representar las frases como el vector promedio de los *embeddings* de las palabras que la componen. Los modelos utilizados fueron *Support Vector Classification*, *Gaussian Naive Bayes*, *Multi-layer Perceptron classifier*, *Random Forest Classifier* y *Logistic Regression*, todos de la librería *sklearn*.

En la figura 4.10 se ilustra los resultados obtenidos con *word embeddings* y *embeddings* polares para distintos niveles de varianza representada y en las figuras 4.11 y 4.12 los resultados de cada *embedding* pero comparado con el número de dimensiones que resulta del análisis de componentes principales.

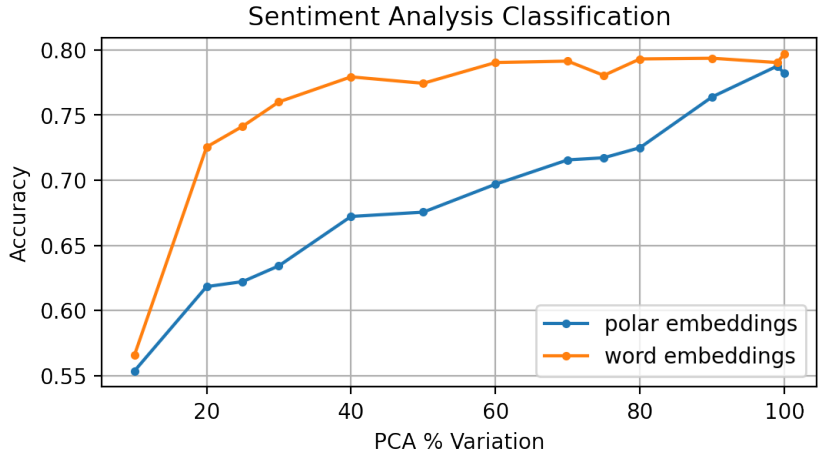


Figura 4.10: Gráfico de *performance* de los *embeddings* reducidos en tarea de análisis de sentimientos.

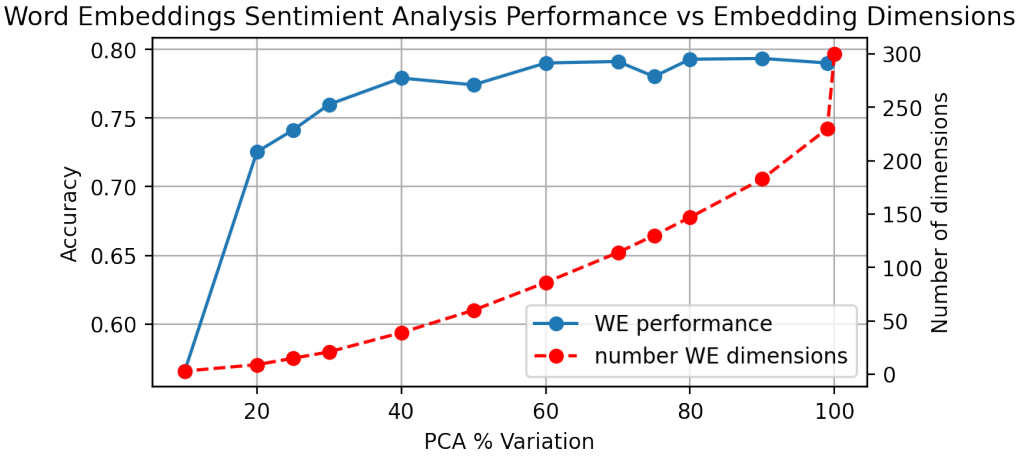


Figura 4.11: Gráfico de *performance* de los *word embeddings* reducidos en tarea de analisis de sentimientos versus las dimensiones de los *word embeddings*.

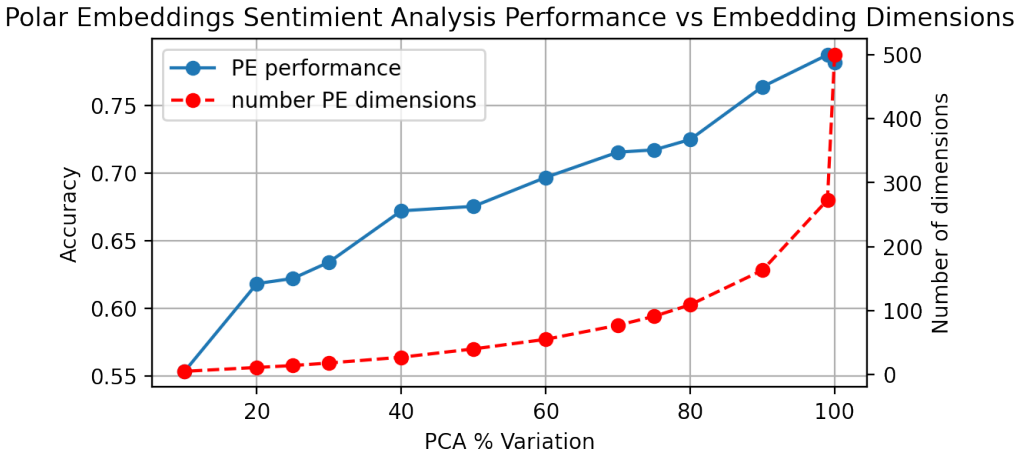


Figura 4.12: Gráfico de *performance* de los *embeddings* polares reducidos en tarea de clasificación de analisis de sentimientos versus las dimensiones de los *embeddings* polares.

Se puede observar de la figura 4.10 que la *performance* de los *word embeddings* no disminuye significativamente hasta bajar del 60 % de representación de varianza. En cambio, baja bastante la *performance* desde el 90 % hacía abajo, sin embargo, tiene la peculiaridad que sube en el paso del 100 % al 99 %, acercándose a los resultados de los *word embeddings*. En los resultados graficados en 4.11 y 4.12 se puede comprobar que el paso al 99 % de varianza representada genera el mayor salto en la disminución de dimensiones, siendo, además, el paso que menos pérdida de *accuracy* tiene para ambos *embeddings*.

Conclusión

En este trabajo se desarrolló un método de cuantificación de complejidad de texto en lenguaje natural, recogiendo supuestos lingüísticos y aprovechando herramientas del procesamiento de lenguaje natural. El método desarrollado consta de modelar las dimensiones de un texto como ejes semánticos, basados en el supuesto que las palabras tienen múltiples propiedades y el significado connotativo que le dan las personas puede ser valorado. Luego, se crean representaciones vectoriales de los textos usando como base *word embeddings* que son transformados a *polar embeddings* y, finalmente, reduciendo la dimensionalidad para cierta varianza representada con análisis de componentes principales. A continuación, el método fue probado y validado en un caso de estudio con un *dataset* de noticias provenientes de la *BBC*. Además, se realizó una evaluación cualitativa para poder ejemplificar lo que el método estaba capturando como complejidad, analizando textos que el método catalogó como de alta complejidad y de baja complejidad; y una evaluación cuantitativa que consistió en ocupar los resultados del método desarrollado en tareas de calificación de noticias y de análisis de sentimientos.

En cuanto al objetivo general de cuantificar la complejidad de un texto en lenguaje natural con la que se puede representar se considera que se cumplió al tener un método que expresa numéricamente la complejidad de un texto, pero de una forma distinta a la que motiva este trabajo de memoria. El *framework* que mide la complejidad de debates de internet con patrones de votaciones entregaba una complejidad que pareciera indicar las dimensiones más significativas de una discusión, en cambio este método, al darle una base lingüística con los ejes semánticos, genera un espectro muy amplio de dimensiones basado en las propiedades con las que se pueden describir las palabras, que al mismo tiempo es primordial para extender el análisis de complejidad a todo tipo de textos en lenguaje natural. Si se desea conocer la cantidad de dimensiones y cuáles son, es el espectro de dimensiones mencionado el que no permite que este método sea tan sistemático como la metodología de los patrones de votaciones, porque depende de los ejes semánticos que se escogen arbitrariamente para modelar las dimensiones y de la varianza representada que se busca abarcar. Sin embargo, este método muestra ser útil como medida de comparación de complejidad: en la comparación de documentos de alta y baja complejidad (como se vio en la sección 4.1) y entre categorías completas de noticias (como se vio en la sección 3.1.3). Es decir, con este método se puede determinar si un documento es más complejo que otro, o si un conjunto de documentos es más complejo que otro. No sólo eso, sino que la capacidad de comparación sí se puede sistematizar porque se vio que existe una correlación entre reducir *embeddings* polares y *word embeddings* (esto se puede observar en las visualizaciones de la sección 3.1.2), por lo que si lo que se busca es comparar complejidad sin interpretabilidad, se puede comparar utilizando la

reducción de *word embeddings*. Y con respecto a la elección arbitraria de la varianza representada, demostró ser una ventaja, ya que es útil como medida de sensibilidad. En tanto una varianza representada alta sirvió para discriminar documentos en más grupos (figura 3.6), una varianza representada más baja sirvió para exponer las diferencias de complejidad entre conjuntos de documentos (ver figura 3.9).

Con respecto a los objetivos específicos, el primero era desarrollar el método para cuantificar la complejidad, que se considera logrado al modelar la solución basada en la reducción de los *embeddings* polares (sección 2.1); el segundo era crear una herramienta que recibe texto en lenguaje natural y retorna la complejidad, que se considera logrado al hacer una implementación del método modelado (sección 2.2); y el tercero era recopilar datos de textos para ser estudiados y hacer una evaluación de resultados, que se considera logrado ya que se llevó a cabo el caso de estudio de noticias de la *BBC* que permitió validar el método en un caso práctico y para una evaluación cualitativa, también ser realizó una evaluación cuantitativa en tareas típicas de procesamiento de lenguaje natural.

La interpretabilidad no estaba contemplada dentro de los objetivos de esta memoria, pero es beneficioso para un método de estas características. Los resultados obtenidos en la evaluación cualitativa 4.1 no parecen ser de gran utilidad, esto se puede deber a la elección de ejes semánticos, que fueron antónimos de la *web* especializada *WordNet*, tal como hace *POLAR*. Si bien utilizar pares de antónimos como en *POLAR*, *SemAxis* y *FrameAxis* le da generalidad a los ejes semánticos para ser utilizados en cualquier corpus, para una búsqueda de interpretabilidad de complejidad no son óptimos. Encontrar ejes semánticos que puedan dar interpretación a las dimensiones depende de las características del corpus y es un problema independiente y que excede los alcances de este trabajo. Al hablar de interpretabilidad cabe mencionar que los resultados de los *loading scores* (los pesos de los componentes principales) no correlacionan con las medidas probadas de *intensity* o *bias* de *FrameAxis* (sección 3.1.5), por lo que no se puede reemplazar el estudio de interpretabilidad con estas técnicas.

Gracias a la evaluación cuantitativa de la sección 4.2 se puede concluir que este método entrega una representación dimensional que puede ser utilizada en tareas de clasificación. Éste es un ejemplo de cómo el método puede ser relevante ya que dentro de ciertos rangos de varianza representada permite mantener la calidad de los resultados disminuyendo la dimensionalidad arbitraria de los *embeddings*.

El trabajo realizado en esta memoria puede ser útil como punto de partida para analizar la complejidad de temáticas sociales actuales o con perspectivas históricas, pero para explotar la interpretabilidad del método se necesita pulir la elección de ejes semánticos. Es por lo anterior que dentro de los trabajos futuros que podrían complementar este método está realizar un estudio de cómo seleccionar los mejores ejes semánticos que puedan brindar una mejor interpretabilidad a un corpus específico. Otro trabajo que se puede estudiar es aprovechar el mismo *dataset* al que se medirá la complejidad para entrenar los *word embeddings* que se utilizarán como base, o, como *SemAxis*, entrenar los *word embeddings* en el corpus que se quiere estudiar junto con un otro de gran tamaño, de tal manera que se aproveche la cantidad de información del corpus grande y que se influya con el contexto semántico del corpus objetivo. Y, por último, se puede trabajar para encontrar la forma de darle importancia al orden de las palabras, ya que en este método se analizan las palabras individualmente y no

importa la sintaxis del texto.

Finalmente, mencionar las lecciones aprendidas entre las que se cuentan la dificultad de construir una solución desde cero sobre todo para un problema tan abstracto, la relevancia de investigar fuente útiles para basar el desarrollo de una solución, la importancia de recoger fuentes de otras disciplinas para enriquecer un planteamiento (en este caso la lingüística) y lo beneficioso que es desarrollar trabajos que puedan ser replicables y facilitar su uso para que en un futuro otras personas puedan aprovechar ese conocimiento.

Bibliografía

[1] Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. SemAxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2450–2461, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[2] Adam Boche, Jeffrey B. Lewis, Aaron Rudkin, and Luke Sonnet. The new voteview.com: preserving and continuing keith poole's infrastructure for scholars, students and observers of congress. *Public Choice*, 176(1-2):17–32, 2018.

[3] Dennis Chong and James N. Druckman. Framing theory. *Annual Review of Political Science*, 10(1):103–126, 2007.

[4] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 377–384, New York, NY, USA, 2006. Association for Computing Machinery.

[5] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas, November 2016. Association for Computational Linguistics.

[6] Haewoon Kwak, Jisun An, Elise Jing, and Yong-Yeol Ahn. Frameaxis: Characterizing microframe bias and intensity with word embedding. *PeerJ Computer Science*, 7:e644, 2021.

[7] Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.

[8] Binny Mathew, Sandipan Sikdar, Florian Lemmerich, and Markus Bernhard Strohmaier. The POLAR Framework : Polar Opposites Enable Interpretability of Pre-Trained Word Embeddings. In *Proceedings of the World Wide Web Conference WWW 2020 / editors: Yennun Huang, Irwin King, Tie-Yan Liu, Maarten van Steen*, pages 1548–1558, New York, NY, Apr 2020. 29th Web Conference, Taipei (Taiwan), 20 Apr 2020 - 24 Apr 2020, Association for Computing Machinery.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[10] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(1):39–41, 1995.

[11] Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. The measurement of meaning. *University of Illinois press*, (47), 1957.

[12] Liton Chandra Paul, Abdulla Al Suman, and Nahid Sultan. Methodological analysis of principal component analysis (pca) method. *International Journal of Computational Engineering & Management*, 16(2):32–38, 2013.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, page 1532–1543, 2014.

[15] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[16] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[17] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *In Thirty-first AAAI conference on artificial intelligence*, page 4444–4451, 2017.

[18] Utkarsh Upadhyay, Abir De, Aasish Pappu, and Manuel Gomez-Rodriguez. On the complexity of opinions and online discussions. In J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman, editors, *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 258–266. ACM, 2019.

ANEXOS

Anexo A

Más Ejemplos de Documentos de Baja Complejidad

Para expander los ejemplos presentados en la sección de evaluación cualitativa 4.1 se incluyen las siguientes noticias de *dataset* de la *BBC* que demuestran los resultados del método en cuánto a calificación como alta o baja complejidad.

El segundo documento menos complejo se titula *Howard hits back at mongrel jibe*, es de la categoría Política y tiene una complejidad de 69 dimensiones. Su contenido es el siguiente:

> *Michael Howard has said a claim by Peter Hain that the Tory leader is acting like an "attack mongrel" shows Labour is "rattle" by the opposition. In an upbeat speech to his party's spring conference in Brighton, he said Labour's campaigning tactics proved the Tories were hitting home. Mr Hain made the claim about Tory tactics in the anti-terror bill debate. "Something tells me that someone, somewhere out there is just a little bit rattled," Mr Howard said. Mr Hain, Leader of the Commons, told BBC Radio Four's Today programme that Mr Howard's stance on the government's anti-terrorism legislation was putting the country at risk. He then accused the Tory Leader of behaving like an "attack mongrel" and "playing opposition for opposition sake". Mr Howard told his party that Labour would "do anything, say anything, claim anything to cling on to office at all costs". "So far this year they have compared me to Fagin, to Shylock and to a flying pig. This morning Peter Hain even called me a mongrel. "I don't know about you, but something tells me that someone, somewhere out there is just a little bit rattled." Environment Secretary Margaret Beckett rejected Mr Howard's comment, telling Radio 4's PM programme that Labour was not "rattled". "We have a very real duty to try to get people to focus on Michael Howard's record, what the proposals are that he is trying to put forward to the country and also the many examples we are seeing now of what we believe is really poor judgement on his behalf." Mr Howard said Tory policies on schools, taxes, immigration and crime were striking a chord with voters. "Since the beginning of this year - election year - we've been making the political weather," he told the party conference. Mr Howard denied he had*

been "playing politics" by raising the case of Margaret Dixon, whose operation had been cancelled seven times, which grabbed headlines for the party two weeks ago. And he hit back at Labour claims he had used Mrs Dixon as a "human shield". "She's not a human shield Mr Blair, she's a human being." Mr Howard said his party plans for immigration quotas, which have also been the focus of much media coverage, were not "racist" - just "common sense". He pledged cleaner hospitals and better school discipline, with a promise to get rid of "political correctness" in the national curriculum and give everyone to the same chance of a "decent" state education as he had. "I come from an ordinary family. If the teenage Michael Howard were applying to Cambridge today, Gordon Brown would love me." And he stressed his party's commitment to cut taxes and red tape and increase the basic state pension in line with earnings. He finished with a personal appeal to party activists to go out and win the next election. "One day you will be able to tell your children and grandchildren as I will tell mine, 'I was there. I did my bit. I played my part. I helped to win that famous election - the election that transformed our country for the better'." Labour election co-ordinator Alan Milburn said: "Michael Howard's speech today confirms what we have always said - that his only strategy is opportunism but he has no forward vision for the country. In reference to the appearance of Mr Howard's family on the conference stage with him, Mr Milburn said: "Michael Howard is perfectly entitled to pose with his family today. "But it is the hard working families across Britain that will be damaged by his plan to cut £35bn from public spending"

El tercer documento menos complejo se titula *BNP leader Nick Griffin arrested*, es de la categoría Política y tiene una complejidad de 70 dimensiones. Su contenido es el siguiente:

The leader of the British National Party has been arrested as part of a police inquiry following the screening of a BBC documentary. A party spokesman said Nick Griffin was arrested on Tuesday morning on suspicion of incitement to commit racial hatred. West Yorkshire police confirmed they had arrested a 45-year-old man from outside their area. BNP founding chairman John Tyndall was arrested on Sunday on the same charge. In July, the BBC documentary Secret Agent featured covertly-filmed footage of BNP activists. Mr Griffin is the twelfth man to be arrested following the documentary. Nine men from West Yorkshire and another man from Leicester have been arrested and freed on bail. Seven of the men had been held variously in connection with suspected racially aggravated public order offences, conspiracy to commit criminal damage and possession of a firearm. Two men, both from Keighley, were arrested in September on suspicion of conspiracy to commit criminal damage. A 24-year-old man from Leicester was detained on Monday on suspicion of incitement to commit racial hatred. A BNP spokesperson said Mr Tyndall, from Brighton, was arrested following a speech he made in Burnley, Lancashire, and was released on police bail.

Anexo B

Más Ejemplos de Documentos de Alta Complejidad

El segundo documento más complejo se titula *GTA sequel is criminally good*, es de la categoría Tecnología y tiene una complejidad de 110 dimensiones. Su contenido es el siguiente:

> *The Grand Theft Auto series of games have set themselves the very highest of standards in recent years, but the newest addition is more than able to live up to an increasingly grand tradition. The 18 certificate GTA: San Andreas for the PlayStation 2 could have got away with merely revisiting a best-selling formula with a more-of-the-same approach. Instead, it builds and expands almost immeasurably upon the last two games and stomps, carefree, over all the Driv3r and True Crime-shaped opposition. Even in the year that will see sequels to Halo and Half-Life, it is hard to envisage anything topping this barnstorming instant classic. The basic gameplay remains familiar. You control a character, on this occasion a youth named CJ, who sets out on a series of self-contained missions within a massive 3D environment. CJ can commandeer any vehicle he stumbles across from a push-bike to a city bus to a plane. All come in handy as he seeks to establish his presence in a tough urban environment and avenge the dreadful deeds waged upon his family. To make things worse, he is framed for murder the moment he arrives in town, and blackmailed by crooked cops played by Samuel L Jackson and Chris Penn. The setting for all this rampant criminality is the fictional US state of San Andreas, comprising three major cities: Los Santos, which is a thinly-disguised Los Angeles, San Fierro, aka San Francisco and Las Venturas, a carbon copy of Las Vegas. San Andreas sucks you in with its sprawling range, cast of characters and incredibly sharp writing. Its ability to capture the ambience of the real-world versions of these cities is something to behold, assisted no end by the monumental graphical advances since Vice City. The streets, and vast swathes of countryside, are by turns gloriously menacing, grungy and preppy. Flaunting awesome levels of graphical detail, the game's overall look, particularly during the many unusual weather conditions and dramatic sunsets, is stupendous. The outstanding bread-and-butter gameplay mechanics provide a solid grounding for the elaborate plot to hang on. Cars handle more convincingly than ever, a superb motion blur kicks in*

when you hit high speeds, and there's more traffic to navigate than before. Park your vehicle across the lanes of a freeway, and within seconds there will be a huge pile-up. Pedestrians are also out in force, and are a loquacious bunch. CJ can interact with them using a simple system on the control pad. They will pass comments on his appearance and credibility, aspects that the player now has control over. Clothes, tattoos and haircuts can all be purchased, and funding these habits can be achieved by criminal means or by indulging in mini-games like betting on horses and challenging bar patrons to games of pool. The character will put on or lose weight according to how long he spends on foot or in the gym. He will have to pause regularly in restaurants to keep energy levels up, but will swell up as a result of over-eating. And at last, this is a GTA hero who can swim. At a time when games are once again under fire for their supposed potential to corrupt the young, San Andreas' violence, or specifically the freedom it gives the player to commit violence, are sure to inflame the pro-censorship brigade. Developers Rockstar have not shied away from brutality, and in some respects ramp it up from past outings. When hijacking a car, for example, CJ will gratuitously shove the driver's head into the steering wheel rather than just fleeing with the vehicle. Indeed, the tone is darker than the jokey Vice City. The grim subject matter here hardly lends itself to gags in quite the same way as the cheesy 80s setting of the last game. This title, incidentally, is set in 1992, but that is really neither here nor there apart from the influence it has on the radio playlists. The wit is still present, just more restrained than in previous outings. A further reason for this is that the incredible range of in-vehicle radio stations available means you will spend less time happening upon the hilarious talk radio options, where GTA games' trademark humour is anchored. The quality of voice acting and motion capture is simply off-the-chart. The game's rather odious gangland lowlifes swagger and mouth off in a way that rings very true indeed. It is a testament to San Andreas' magnificence that it has a number of prominent flaws, but plus-points are so numerous that the niggles don't detract. The on-screen map, for instance, is needlessly fiddly, an unwelcome change from past editions. There is also a very jarring slowdown at action-packed moments. And the game suffers from the age-old problem that can be relied upon to blight all games of this genre, setting you back a vast distance when you fail right at the very end of a long mission. But the gameplay experience in its entirety is overwhelmingly positive. You simply will not be bothered by these minor failings. San Andreas is among the few unmissable games of 2004.

El tercer documento más complejo se titula *2D Metal Slug offers retro fun*, es de la categoría Tecnología y tiene una complejidad de 108 dimensiones. Su contenido es el siguiente:

Like some drill sergeant from the past, Metal Slug 3 is a wake-up call to today's gamers molly-coddled with slick visuals and fancy trimmings. With its hand-animated sprites and 2D side-scrolling, this was even considered retro when released in arcades four years ago. But a more frantic shooter you will not find at the end of your joypad this year. And yes, that includes Halo 2. Simply choose your grunt and wade through five 2D side-scrolling levels of the most hectic video game blasting you will ever encounter. It is also the toughest game you are likely to play, as hordes of enemies and few lives pile the pressure on. Players must battle

soldiers, snowmen, zombies, giant crabs and aliens, not to mention the huge, screen-filling bosses that guard each of the five levels. The shoot-anything-that-moves gameplay is peppered with moments of old-school genius. Fans of robotic gastropods should note the title refers, instead, to the vast array of vehicles on offer in a game stuffed with bizarre hardware. Tanks, jets and submarines can be commandeered, as well as cannon-toting camels, elephants and ostriches - more weaponry on offer than in an acre of Iraq. Doling out justice is a joy thanks to ultra responsive controls, and while this is a tough nut to crack, it is addictive enough to have you gagging for that one last go. And at a mere £20, Metal Slug 3 is as cheap as sliced, fried spuds, as the man says. Of course, most of you will ignore this, lacking as it does the visual fireworks of modern blasters. But at a time when blockbuster titles offer only a fresh lick of paint in favour of real innovation, Metal Slug 3 is a fresh gasp of air from an era when the Xbox was not even a twinkle in Bill Gates' eye.