Identificación y relacionamiento de elementos comunes en testimonios escritos de víctimas del conflicto armado en Colombia usando inteligencia artificial

Fabián Orlando Baena Henao

Tesis de maestría presentada para optar al título de Magíster en Ingeniería

Directores

Yony Fernando Ceballos, Doctor (PhD) en Ingeniería

Julián Andrés Castillo Grisales, Magíster (MSc) en Ingeniería

Asesora

Ana María Tangarife Patiño, Doctor (PhD) en Traducción y Ciencias del Lenguaje

Universidad de Antioquia

Facultad de Ingeniería

Maestría en Ingeniería

Medellín, Antioquia, Colombia

2024

Cita	(Baena Henao, 2024)
Referencia **Estilo APA 7 (2020)**	Baena Henao, F. O. (2024). *Identificación y relacionamiento de elementos comunes en testimonios escritos de víctimas del conflicto armado en Colombia usando inteligencia artificial* [Tesis de maestría]. Universidad de Antioquia, Medellín, Colombia.

Maestría en Ingeniería.

Grupo de Investigación Ingeniería y Sociedad (I&S).

Centro de Investigación Ambientales y de Ingeniería (CIA).

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Dedicatoria

A mí.
Todos los días sale el sol.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis directores de trabajo de grado por su confianza, ya que a pesar de provenir de un área en la que hay muchos más matices y colores que fórmulas y algoritmos, sus enseñanzas, más allá de tener aplicaciones ingenieriles, tuvieron un profundo impacto humano. Al profesor Fernando Ceballos, a quien respeto, admiro y agradezco por su apoyo desde el comienzo, incluso en los momentos en que la incertidumbre me nubló.

Agradezco a mi amiga, la profesora Ana María Tangarife, quien me mostró y me animó a emprender este camino tan anhelado, pero a la vez desconocido.

A todas las personas que confiaron en mí y me alentaron en esta apasionante aventura.

A mi hijo, mi esposa y mi familia, por su compañía e incondicional apoyo.

Tabla de contenido

Lista de tablas

Lista de figuras

Siglas, acrónimos y abreviaturas

CEV	Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición
CNMH	Centro Nacional de Memoria Histórica
FARC-EP	Fuerzas Armadas Revolucionarias de Colombia – Ejército del Pueblo
JEP	Jurisdicción Especial para la Paz
ML	*Machine Learning* – Aprendizaje Automático
PLN	Procesamiento de Lenguaje Natural
SIVJRNR	Sistema Integral de Verdad, Justicia, Reparación y No Repetición
UBPD	Unidad de Búsqueda de Personas dadas por Desaparecidas

Resumen

Tras más de 60 años de violencia en Colombia, originada por la lucha política entre el Estado y diferentes actores, se cuenta con una amplia fuente de testimonios escritos de las víctimas. Realizar un análisis tradicional de este acervo supone una gran inversión en términos de tiempo y recursos, por lo que este trabajo de investigación pretende plantear la aplicación de un método en el que, a través de un conjunto de técnicas de *machine learning*, en las que intervienen principalmente el procesamiento de lenguaje natural, la minería de textos y la similitud de documentos, permita establecer un relacionamiento entre los testimonios de las víctimas, encontrando en ellos elementos comunes, para tratar de comprender temas como el entendimiento y la configuración de la guerra y sus formas, así como la representación y participación de actores y eventos.

Palabras clave: Procesamiento de lenguaje natural, Inteligencia artificial, Similitud de textos, Relacionamiento de textos, Testimonios de víctimas, Análisis de corpus, Corpus anotados.

Abstract

After more than 60 years of violence in Colombia started by the political dispute between the Colombian State and others illegal armed groups, there are a lot of written testimonies from the victims. Perform a traditional analysis of this collection texts require a great investment in time and other resources, so the aim of this work is to apply a method that involve a set of machine learning techniques, like Natural Language Processing, Text Mining and Text Similarity that allow a relationship between the testimonies of the victims, to find common elements, to try to answer issues like the understanding and setting of the war and its forms, as well like the role played by actors and their participation in that events.

Keywords: Natural language processing, Artificial intelligence, Text similarity, Text relationship, Victim testimonies, Corpus analysis, Annotated corpora.

Introducción

Establecer un relacionamiento entre documentos, o procesos de similitud de documentos como puede ser conocido en la lingüística computacional, es una actividad que reviste un alto grado de complejidad, así como un ejercicio costoso en términos de tiempo y recursos físicos, tecnológicos y humanos, pues implica la conjunción de diferentes elementos y actores de carácter específico, ya que es preciso desarrollar tareas minuciosas como la consolidación y clasificación de textos comunes; posteriormente, un experto en el tema es el encargado de realizar el análisis de cada documento, lo que demanda gran inversión de tiempo en la que influyen variables como la extensión de los textos, así como el carácter técnico y otros elementos cambiantes. Finalmente, exponer los resultados representa, de manera indirecta, subjetividades que dependen del investigador que realiza el análisis.

Este trabajo pretende la aplicación de un método para establecer el relacionamiento automatizado entre testimonios escritos de las víctimas del conflicto armado en Colombia, utilizando técnicas de inteligencia artificial a través del procesamiento de lenguaje natural (PLN), como los algoritmos de similitud de textos.

Para alcanzar el objetivo, la metodología se divide en 5 etapas: **1)** Inicialmente, se consolida un *corpus* de textos a partir de los relatos escritos de las víctimas del conflicto armado, cuya fuente principal proviene de los testimonios recogidos por entidades como la Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición (CEV), el Centro Nacional de Memoria Histórica (CNMH), entre otros. **2)** Debido a que los documento no cuentan con un formato estandarizado, la segunda etapa consiste en la normalización del *corpus*, tanto en formato, forma y contenido del texto. **3)** La tercera etapa corresponde a la identificación de las dimensiones, que servirán para establecer las relaciones. **4)** La cuarta etapa aborda la aplicación de los algoritmos de procesamiento de lenguaje natural para identificar patrones comunes entre los documentos, establecer relaciones y distancia entre ellos. **5)** En la última etapa se muestran las relaciones en grafos, con la intención de identificar visualmente las similitudes y diferencias que arrojen los algoritmos.

Encontrar patrones comunes entre documentos y establecer relaciones con un método estándar, es un ejercicio que aporta a un mejor entendimiento de los datos, pues permite descubrir elementos ocultos o realizar interpretaciones, dependiendo de los contextos en los que sean analizados.

1 Justificación

Tras más de 60 años de violencia política en Colombia, propiciada principalmente por el conflicto bélico entre la guerrilla de las FARC-EP y el Estado colombiano, al que posteriormente se sumaron nuevos grupos guerrilleros, pero también, otros actores como el paramilitarismo y el narcotráfico, comienzan a evidenciarse múltiples hechos violentos, tales como secuestros, violaciones, desapariciones, despojos y destierro, que dejaron una huella indeleble en quienes los padecieron: las víctimas.

De otro lado, tras la consolidación del *Acuerdo de Paz* entre las FARC-EP y el Gobierno de Colombia, mediante el acuerdo final para finalizar el conflicto, se crea el Sistema Integral de Verdad, Justicia, Reparación y No Repetición (SIVJRNR), sustentado en el Acto Legislativo 01 (Colombia. Congreso de la República, 2017) y el Decreto 588 (Colombia. Presidencia de la República, 2017), con el propósito de conocer la verdad de lo ocurrido en el marco del conflicto armado y contribuir al esclarecimiento de las diferentes formas de violencia, así como ofrecer una explicación amplia a toda la sociedad (Colombia. Comisión de la Verdad, 2024b).

A fin de alcanzar el objetivo para el cual fue creado, este sistema cuenta entre sus mecanismos con la Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición (CEV), cuyos fines principales son el esclarecimiento del conflicto armado, identificando patrones y causas: el derecho de las víctimas a la verdad, el reconocimiento de lo sucedido durante la guerra y sentar las bases para la no repetición (Colombia. Presidencia de la República, 2017). Para conseguir el objetivo, la CEV recogió vastas cantidades de testimonios, representadas en múltiples voces, formas y formatos, entre las que se encuentran principalmente las de las víctimas, pero también de los victimarios.

Si bien el acervo de testimonios se configura como elemento fundamental para cumplir con los objetivos de la Comisión a través de su análisis e interpretación, el gran volumen que este representa da a entender que se puede recuperar aún más información oculta entre líneas, o que cierta información no considerada por la CEV en sus informes represente relevancia para otros, así

como las diversas interpretaciones e intereses que pueda representar para demás personas e instituciones.

Por otra parte, los avances para el manejo de grandes cantidades de información (no solo numérica, sino también en formatos de texto, imágenes y video, representados tanto en la aplicación de métodos estadísticos como en automatizaciones de procesos a través de algoritmos computacionales) han aportado al trabajo que científicos de las ciencias sociales e ingenieriles realizan al análisis de *corpus* de textos con herramientas ágiles y robustas, pues de manera tradicional los procesos de análisis de datos de texto se han considerado como una costosa empresa en términos de tiempo y recursos tecnológicos y humanos.

Considerando lo expuesto, este trabajo pretende establecer una metodología para el análisis automático de testimonios, consolidando un *corpus* a partir de los relatos escritos de las víctimas del conflicto armado en Colombia, transformando los datos para su interpretación y aplicando un método que permita encontrar y graficar elementos comunes entre ellos. De igual forma, los resultados de este trabajo se enmarcan en la consolidación de herramientas, metodologías y procedimientos que apuntan a la interpretación de textos y relatos escritos relacionados con las vivencias de las víctimas del conflicto armado en Colombia. Asimismo, se perfila como una herramienta que facilita el entendimiento de la configuración de la guerra en nuestro país, los actores, sus roles y participación; pero también, como medio facilitador de los fines enmarcados en los procesos de posconflicto considerados en los mecanismos del SIVJRNR, tales como la CEV, la Jurisdicción Especial para la Paz (JEP), La Unidad de Búsqueda de Personas dadas por Desaparecidas (UBPD), las medidas de reparación integral para la construcción de paz y garantías de no repetición. Adicionalmente, el beneficio de apoyo en la interpretación de escritos de este tipo que se proyecte en distintas instituciones y particulares.

2 Planteamiento del problema

Desde finales de la década de los años 50, Colombia ha estado inmersa en una violencia comprendida, entre otras causas, por la lucha política entre el Estado y las guerrillas, que se fue agudizando conforme se fueron sumando otros actores bélicos tales como grupos paramilitares, narcotráfico e incluso el mismo Estado. Se estima que entre 1958 y 2012 el conflicto armado en Colombia dejó por lo menos 220.000 muertos (Colombia. Centro Nacional de Memoria Histórica. Grupo de Memoria Histórica, 2013).

Los efectos persistentes de la guerra en Colombia han contribuido a la victimización de otro número considerablemente alto de personas, manifestadas en diversas formas como la desaparición forzada, la violencia sexual, el reclutamiento de menores y el desplazamiento forzado, que, según datos tomados a 2013 del Registro Único de Víctimas, asciende a más de 4.744.046 personas (Colombia. Centro Nacional de Memoria Histórica. Grupo de Memoria Histórica, 2013). Estas cifras y formas pueden aumentar, dado que muchos datos se encuentran en subregistros no oficiales o no reportados.

Las víctimas del conflicto, quienes son definidas en el Artículo 5 de la Ley 975 (Colombia. Congreso de la República, 2005) como un sujeto violentado y con derecho a ser reparado, amparadas en el marco de la Comisión de la Verdad, cuyos fines son el derecho a la verdad y a la reparación, así como evitar la repetición (Colombia. Comisión de la Verdad, 2022, 2024b), han querido documentar sus vivencias desde la memoria, permitiendo así la posibilidad de esclarecer hechos, motivos, intereses, así como actores directos e indirectos. Esta memoria, que recoge testimonios de los actores más vulnerables del conflicto tales como niños, mujeres y adultos mayores, representa las diferentes formas de ver y expresar sus emociones, recuerdos, vivencias y cicatrices del conflicto. Tales formas de la memoria son tan variadas tanto como sus formatos de representación, puesto que obras de teatro, exposiciones, narraciones y otros, se recogen en documentos audiovisuales, orales y escritos que dan cuenta del conflicto mismo y de las formas de resistencia a estos hechos (Colombia. Centro Nacional de Memoria Histórica. Grupo de Memoria Histórica, 2013).

Las experiencias recopiladas de las víctimas (de manera particular los testimonios escritos) contienen gran cantidad de información a través de patrones clave de espacio, tiempo, género, individuos, grupos, entre muchos otros, que al ser identificados es posible establecer conexiones que pueden ayudar a entender la configuración del conflicto, tales como sucesos, motivos o modos, solo por mencionar algunos, además de facilitar el esclarecimiento de la verdad y evitar la repetición. Por lo tanto, procesar cada testimonio implica un considerable esfuerzo en términos de recursos y tiempo, dado que para establecer relaciones entre ellos se requiere el trabajo de investigadores de distintas disciplinas, entre los que destacan, solo por mencionar algunos, antropólogos, archivistas e historiadores; de igual manera, adicional a las ciencias sociales, participan investigadores de áreas diferentes, quienes aportan perspectivas y metodologías diversas para enriquecer el análisis. Estos profesionales los interpretan, identificando elementos representados a través de frecuencias de palabras con sentido semántico dentro de un *corpus* (Colombia. Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición, 2022), pues este último término consiste en focalizar datos observables a modo de evidencia científica (Parodi, 2008).

Realizar un análisis y relacionamiento entre testimonios de víctimas es un proceso que reviste gran importancia, tanto para investigadores de las ciencias sociales interesados en identificar elementos comunes en los testimonios de las víctimas de la violencia en Colombia, como para diversas organizaciones, entre ellas la Comisión de la Verdad, la cual enuncia sus cuatro objetivos: Esclarecimiento, Reconocimiento, Convivencia y No repetición; propende por «conocer la verdad de lo ocurrido en el marco del conflicto armado y contribuir al esclarecimiento de las violaciones e infracciones cometidas durante el mismo y ofrecer una explicación amplia de su complejidad a toda la sociedad» (Colombia. Comisión de la Verdad, 2024b), y que, para lograrlo, en el primer objetivo de escucha para el esclarecimiento, entre el año 2018 y 2022, se recogieron 14.971 entrevistas individuales y colectivas de 28.580 personas, las cuales aportaron a la construcción del informe final, y se encuentran disponibles para su consulta en diferentes formatos (Colombia. Comisión de la Verdad, 2024b).

Así pues, cuando el investigador establece un relacionamiento entre los testimonios es un ejercicio clave en su labor. Sin embargo, realizarlo de manera tradicional, es decir, haciendo una

lectura individual de cada texto de testimonios, se convierte en una tarea dispendiosa cuando se presentan grandes volúmenes de información, pues la misma no se encuentra etiquetada ni organizada, y por tanto, al tornarse complejas estas acciones, se hace imperativo encontrar metodologías y estrategias que automaticen los procesos, garantizando efectividad en los resultados y que se reduzcan considerablemente los tiempos y costos.

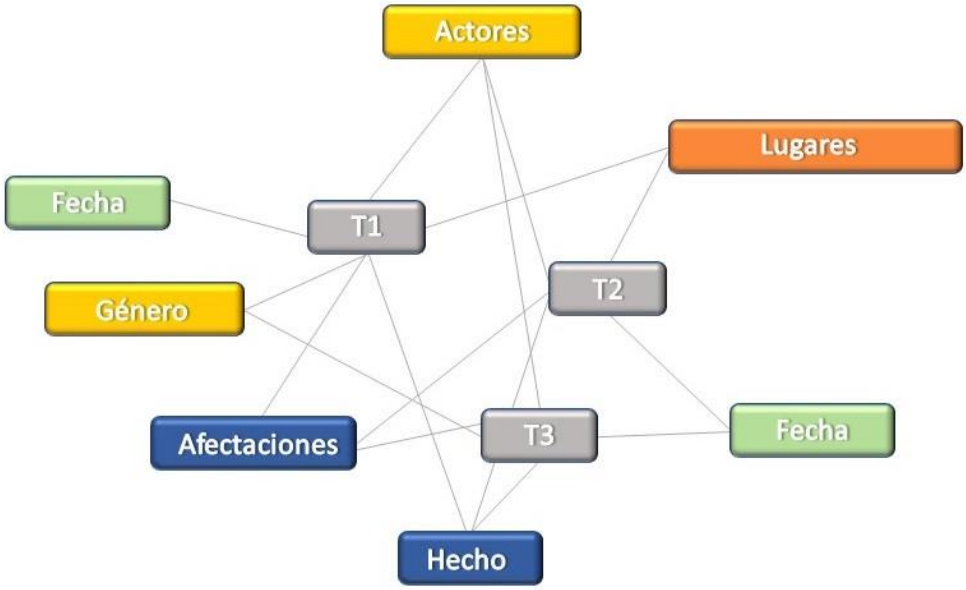
Es en este punto donde la ingeniería y las ciencias sociales convergen, proponiendo metodologías para dar solución a problemas en el estudio de fenómenos sociales, tales como el procesamiento de lenguaje natural a través del uso de algoritmos (Montes y Gómez, 2005), y así lograr la identificación de elementos representativos en los textos, facilitando la recuperación de información y la construcción de nuevo conocimiento (Duque Bedoya, 2009). De igual forma, técnicas de inteligencia artificial como el procesamiento de textos a través de la minería de datos, también son relevantes en el diseño de una metodología a proponer, dado que facilitan, entre otras, la identificación de frecuencias de palabras y las relaciones semánticas dentro del *corpus* (Godoy Viera, 2017), tareas que permiten ejecutar un procesamiento automatizado de información y que supone gran utilidad en los procesos de investigación social que usan testimonios como fuente.

Por todo lo anterior, y considerando las crecientes fuentes de información, así como los mismos datos (representados en testimonios de víctimas del conflicto armado colombiano), las necesidades de los investigadores en el procesamiento de grandes volúmenes de datos y la disponibilidad de técnicas, modelos y metodologías computacionales para el procesamiento automatizado del lenguaje natural, se propone dar respuesta a la pregunta: ¿qué enfoque podría utilizarse para identificar y relacionar automáticamente los elementos comunes presentes en los textos de los testimonios de las víctimas del conflicto armado en Colombia, con el fin de facilitar el análisis y la comprensión de patrones y temas recurrentes?.

Así que, como ilustración, se presenta la **Figura 1**, con 3 textos asociados con la letra T y acompañados del número del documento, que en este trabajo corresponden a testimonios de víctimas del conflicto armado. De cada documento se desprenden elementos que corresponden a las dimensiones de interés del investigador y que, al ser comunes entre cada documento, es posible determinar que existe una relación.

Figura 1

Relacionamiento de elementos comunes en tres testimonios (T) de víctimas



3 Objetivos

3.1 Objetivo general

Identificar y relacionar elementos comunes en testimonios escritos de víctimas del conflicto armado en Colombia usando inteligencia artificial.

3.2 Objetivos específicos

- Establecer un contexto de aplicación de una metodología que responda a la forma tradicional de comunicación de víctimas del conflicto armado.

- Consolidar un *corpus* de testimonios escritos de víctimas del conflicto armado en Colombia.

- Definir dimensiones o categorías para identificar los elementos comunes y establecer las relaciones entre los testimonios escritos.

- Desarrollar un algoritmo que utilice técnicas de procesamiento de lenguaje natural que permita identificar elementos comunes y su posterior relacionamiento entre los testimonios del *corpus*.

4 Marco teórico

El uso de técnicas de Procesamiento de Lenguaje Natural (PLN, o NLP por sus siglas en inglés *Natural Language Processing*), siendo una rama de la inteligencia artificial que se enfoca en la interacción entre máquinas y el lenguaje humano (Bird et al., 2009), vienen siendo ampliamente utilizadas en aplicaciones de diferentes tipos, entre las que se destacan la generación de resúmenes automáticos, el reconocimiento de entidades, la clasificación de textos, entre muchos otros, y que han aportado a la consecución de tareas generalmente complejas como la identificación del plagio en textos académicos o la generación automática de grafos o mapas conceptuales, solo por mencionar algunas.

Conceptos clave para el entendimiento de estas técnicas y aplicaciones son el procesamiento de textos, el cual se enfoca, principalmente, en la manipulación y el análisis de conjuntos de datos de texto (Hogenboom et al., 2016), desarrollando tareas como limpieza de texto, eliminación de caracteres especiales, tokenización para dividir el texto en unidades más pequeñas (llamadas tokens) en palabras o frases, la corrección ortográfica de palabras, la transformación de caracteres entre mayúsculas y minúsculas, entre otras. Asimismo, representar palabras en vectores numéricos se convierte, tal vez, en la tarea principal del procesamiento del lenguaje natural, por lo que cada palabra del *corpus* representa a un vector en un espacio n-dimensional, donde n es la dimensión de la representación vectorial (Jurafsky & Martin, 2014). Esta técnica facilita la precisión en los algoritmos, al trabajar diferentes tipos de transformaciones a las palabras de los documentos.

Es clave, además, determinar que el PLN puede ser abordado desde enfoques supervisados, no supervisados y la combinación entre ambos, pues el primero referencia un modelo que se entrena utilizando un conjunto de datos etiquetados (Bishop, 2006), es decir, datos para los cuales se conoce la respuesta correcta a fin de predecir un resultado de manera efectiva, considerando los datos de entrenamiento etiquetados previamente; mientras que el segundo, o algoritmo no supervisado, se trata de un enfoque de aprendizaje automático en el que el modelo se entrena utilizando un conjunto de datos no etiquetados (Sumathi & Sivanandam, 2006). Dicho de otra forma, datos para los cuales

no se conoce una respuesta. La principal intención de esta técnica es la identificación de patrones, tales como grupos, clústeres o relaciones, sin contar con etiquetas asociadas.

Por su parte, y como punto de interés relevante para esta investigación, la similitud de textos o de documentos se destaca como elemento fundamental en la aplicación del PLN, pues esta (la similitud de documentos) es concebida como el proceso de utilizar una métrica basada en distancia o similitud que puede identificar qué tan similar es un documento de texto a cualquier otro en función de características extraídas (Sarkar, 2019). Esta técnica ha sido implementada para resolver diversos problemas, entre los que se destacan la búsqueda de información en bases de datos, la detección de duplicados o plagio (Mariani et al., 2018), la clasificación de documentos (Ko et al., 2012) y la recomendación de contenido (Sugiyama & Kan, 2015). Con relación a las métricas para medir la similitud entre textos o documentos, resaltan:

a) La distancia de coseno: es útil para medir la similitud entre dos vectores de características, por lo que es usada principalmente en la comparación de textos en función de su contenido semántico (Manning et al., 2008). Esta distancia varía entre el rango de -1 y 1. Cuando la métrica obtiene un valor de 1, indica que los vectores son idénticos en dirección, es decir, los textos son completamente similares; por su parte, un valor de -1 es el resultado que demuestra que los textos son completamente opuestos. Si se obtiene un resultado de 0, es indicador de que los textos son completamente ortogonales o no tienen similitud en absoluto.

b) La similitud de Jaccard: corresponde a una métrica utilizada para medir la similitud entre dos conjuntos de palabras o de términos en los documentos en base al cálculo de dividir la intersección y la unión de los elementos en dos conjuntos (Leskovec et al., 2014). Esta métrica muestra como resultado un valor entre 0 y 1, donde 0 significa que no hay elementos en común y 1 que los conjuntos son idénticos.

c) La distancia de Levenshtein: también conocida como distancia de edición, corresponde a una métrica que calcula la similitud entre dos cadenas de caracteres. Esta distancia se basa en el número mínimo de operaciones de edición necesarias para transformar una cadena en otra (Leskovec et al., 2014). El valor resultante es un número entero que calcula el número mínimo de

operaciones, o cambios necesarios, para transformar una cadena en otra. Entre menor sea el valor de esta métrica, la similitud aumenta entre los textos.

Tradicionalmente, la similitud de textos a través del procesamiento de lenguaje natural utiliza técnicas tales como la bolsa de palabras, también conocida como «BoW» o *bag of words*, por sus siglas en inglés, que se utiliza para representar documentos de texto como vectores numéricos, descomponiendo un texto en cada palabra del mismo (Sarkar, 2019). Cada documento se representa como un vector que contiene información sobre la frecuencia de las palabras del documento. Esta técnica se desarrolla considerando un conjunto de actividades, tales como: **a)** la generación de tokens, **b)** la construcción de un vocabulario o vectorización de las palabras, **c)** la representación del documento en dimensiones y frecuencias, para finalmente, **d)** determinar la similitud a partir de una medida de distancia.

Similar a la técnica anterior, la literatura referencia el modelo bolsa de n-gramas o *bag of n-grams*, el cual se diferencia de BoW en que la actual considera no solo una palabra del documento, sino una secuencia contigua de ellas. La ventaja del uso de n-gramas radica en que pueden ser capaces de capturar la estructura y el contexto en un documento de texto (Sarkar, 2019).

Otra de las técnicas tradicionales más utilizas para el análisis de textos es la frecuencia de término - frecuencia inversa de documento, comúnmente conocida por las siglas TF-IDF, la cual es una técnica que asigna pesos a las palabras en función de su frecuencia, la relación con el documento y su importancia en el *corpus* de documentos (Sarkar, 2019). Es utilizada para representar documentos y medir su similitud. Esta técnica se basa en la combinación de dos tareas: **a)** la frecuencia de términos (TF), la cual captura la frecuencia de un vector en un documento, por lo que a medida que cada término es más común dentro del documento, mayor es su valor; y **b)** la frecuencia inversa de documentos (IDF), que mide que la relevancia de un vector dentro de un *corpus* o conjunto documentos, entregando un valor IDF bajo si el término es común en muchos documentos, o alto si es raro o escaso en dicho conjunto. Esta técnica se usa ampliamente en la puntuación de resultados de bases de datos relacionados con las consultas de los usuarios y que gracias a su funcionalidad es adoptada en la recuperación de información y extracción de características de texto. En síntesis, la técnica TF-IDF entrega valores altos a los términos que son

frecuentes en un documento, pero escasos dentro del *corpus*, siendo considerablemente útil en la representación y comparación de documentos en función del contenido.

De igual forma, se identifican técnicas más complejas, como las incrustaciones de palabras o *word embeddings*, la cual es comúnmente utilizada para representar palabras y textos en un espacio vectorial continuo. La idea principal de esta técnica es asignar a cada palabra un vector de números reales, de tal manera que las palabras con significados similares tengan representaciones comunes en el espacio vectorial, lo que apunta a que sea una técnica que considera la semántica para su análisis, logrando que palabras con significados similares se encuentren más cerca una de la otra. Esta técnica se caracteriza por contar con *corpus* etiquetados, tales como Word2Vec (Mikolov et al., 2013) y GloVe (Pennington et al., 2014), teniendo como características la aplicación en algoritmos supervisados y la reducción de la dimensionalidad de los datos que facilita el procesamiento y la comparación de textos.

A su vez, en la literatura se logra identificar la incrustación de documentos, comúnmente conocida como *document embeddings*, siendo similar a la técnica anterior, aunque utilizada para representar documentos completos como vectores a través del promedio de las incrustaciones de palabras en el documento (Le & Mikolov, 2013).

Más recientemente, técnicas y modelos basados en la arquitectura de *transformers*, como lo son BERT (*Bidirectional Encoder Representations from Transformers*) y GPT (*Generative Pre-trained Transformer*) vienen siendo aplicadas al análisis de textos, ya que poseen características como la consideración del contexto y la semántica, siendo técnicas extremadamente útiles en la similitud de documentos. BERT es un modelo de lenguaje pre-entrenado que captura representaciones contextuales de las palabras y los textos, y puede ser utilizado en la medición de la similitud semántica entre dos documentos. Al tratarse de un modelo bidireccional, BERT considera el contexto tanto antes como después de cada palabra en una oración, lo que le permite comprender mejor el significado de las palabras en su contexto (Devlin et al., 2018). El cálculo de la similitud entre los vectores se realiza a través de medidas de distancia como la distancia coseno. Por otro lado, GPT corresponde a un modelo entrenado para generar texto, siendo una de sus aplicaciones más utilizadas, pues se caracteriza por la precisión semántica y de contexto. A

diferencia del modelo anterior, GPT es concebido como un modelo unidireccional, por lo que este solo considera el contexto precedente a una palabra en cada oración (Brown et al., 2020).

Son múltiples y variadas las aplicaciones que la similitud de textos ha abordado basadas en técnicas de *machine learning* y representaciones vectoriales, entre las que se pueden listar:

a) Detección del plagio: esta aplicación es comúnmente utilizada en la identificación de similitudes sustanciales entre documentos, pues su intención es determinar si parte del texto de un documento es igual o muy similar a contenido de otro, en ese caso, categorizándose como plagio. Para determinar esta categorización es preciso realizar la comparación de un documento con un grupo de documentos de referencia. En este sentido, se aborda la detección del plagio de un documento en idiomas diferentes al que se analiza, a fin de considerar la identificación de las mismas ideas en otras lenguas (Barrón-Cedeño et al., 2010). De otro lado, se analiza la detección del plagio utilizando técnicas de redes neuronales convolucionales y modelos entrenados como GloVe y word2vec (Mahmoud & Zrigui, 2020).

b) Recuperación de información: esta aplicación está orientada a la búsqueda y recuperación de documentos e información en la web, motores de bases de datos académicas, repositorios digitales, sistemas de recomendación de contenidos, chatbot y bases de datos de preguntas y respuestas. Se basa en la selección de documentos relevantes a una consulta dada su similitud. Un estudio aborda la selección de documentos de la Red de Repositorios de Ecuador basados en parámetros dados (Vallejo-Huanga et al., 2023).

c) Agrupación de documentos: esta aplicación se basa en la organización automática de documentos en grupos o clústeres según su similitud textual, dadas características comunes o temas similares en un mismo clúster. Esta aplicación es implementada en organizaciones que requieren de la gestión de grandes colecciones de documentos, el análisis ágil y automático de contenido para realizar categorización automática, la identificación de patrones ocultos y temas emergentes en textos no etiquetados. Un estudio analiza el agrupamiento de opiniones de usuarios de la red social X (anteriormente Twitter) con relación a los comportamientos de estacionamiento de vehículos, lo

que permite identificar la similitud, generar grafos para analizar y representar los grupos encontrados (Arhab et al., 2022).

d) Recomendación de contenido: la aplicación de esta técnica es bastante común en diversas plataformas y servicios, como sistemas de recomendación de películas, bases de datos, recomendadores de noticias, sistemas de recomendación de productos en línea, solo por mencionar algunos. Su funcionalidad se basa en la sugerencia de contenido similar a los intereses que el usuario ha demostrado. Un artículo de investigación presenta un recomendador cruzado entre libros y películas, utilizando las técnicas TF-IDF, agrupamiento por K-medias y jerárquico y midiendo la similitud a través de la distancia de coseno (Nawar et al., 2021).

e) Análisis de sentimientos: esta aplicación es usada generalmente para el monitoreo de redes sociales, con el fin de medir la intención de un texto, la afinidad de un candidato o decisión de voto, comparar popularidad de un producto o comentario. El objetivo es determinar qué tan cercano está el comentario o documento de una posición positiva, negativa o neutra. El estudio realizado por Srinivasarao y Sharaff (2023) aborda la aplicación de esta técnica en cuanto al mejoramiento de recomendación de *spam* del correo electrónico a partir del análisis de sentimientos de los mensajes. El sistema que se propone incluye técnicas de extracción de características como TF-IDF.

Por otro lado, con relación a la evaluación de la precisión de los resultados de los modelos antes mencionados, se han utilizado técnicas para medir la similitud entre textos (Gomaa & Fahmy, 2013), tales como:

a) la evaluación basada en conjuntos de datos anotados, la cual consta de un conjunto de documentos etiquetados con sus niveles de similitud y su posterior comparación con las medidas entregadas por el algoritmo; en otras palabras, en esta técnica, la precisión puede ser calculada por la comparación entre las medidas de similitud que realice un experto y las que arroje el algoritmo.

b) la evaluación mediante tareas de clasificación, la cual consiste en clasificar textos como similares o no, para luego compararlos con métricas de evaluación de clasificación. En esta técnica, es posible calcular la precisión determinando en que porcentaje el algoritmo logra clasificar correctamente las inicialmente evaluadas.

c) la evaluación basada en la recuperación de información, la cual se usa para conocer si los documentos presentados por una consulta son relevantes. Esta técnica se usa en consultas de información, en las que, tras obtener los datos, se determina cuantos son relevantes y así determinar la precisión.

d) la validación cruzada, la cual utiliza técnicas mucho más detalladas en divisiones de datos como la *k-fold.* En esta técnica se dividen los datos en múltiples partes, se entrena y prueba el modelo varias veces, y promedia los resultados para obtener una evaluación más robusta y confiable del rendimiento del modelo.

e) evaluaciones con pruebas estadísticas, cuyo objetivo es determinar si existen diferencias a través de la comprobación de hipótesis. En esta técnica, la prueba T, se utiliza para comprobar hipótesis de evaluación.

f) evaluaciones de aplicaciones prácticas pilotos, las cuales consisten en obtener recomendaciones de expertos basadas en eventos reales.

Por su parte, las métricas más utilizadas en las mencionadas técnicas son *precision* (precisión), *recall* (recuperación o sensibilidad) y F1-Score (Goutte & Gaussier, 2005). Por un lado, *precision* pretende determinar el porcentaje de documentos realmente correctos entre el total de predicciones dadas como positivas en el que están incluidas tanto las predicciones de verdaderos positivos como falsos negativos. La fórmula para calcular está metrica está representada de la siguiente manera:

$$\text{Pr}\,e\,cision\ = \frac{Verdaderos\ positivos\ (TP)}{Verdaderos\ positivos\ (TP)\ +\ Falsos\ positivos\ (FP)}$$

El *recall*, por su parte, intenta medir la capacidad que tiene el modelo de identificar la mayor cantidad de documentos correctos del total de documentos que son realmente correctos o verdaderos positivos, siendo el indicador que determina que tan bien el modelo puede recuperar todos los documentos positivos.

Esta métrica está representada en la siguiente fórmula:

$$Recall = \frac{Verdaderos\ positivos\ (TP)}{Verdaderos\ positivos\ (TP) + Falsos\ negativos\ (FN)}$$

Mientras tanto, el F1- Score corresponde a la métrica que considera las dos técnicas anteriores (*precision* y *recall*) para entregar una única medida de rendimiento, estableciendo un equilibrio entre verdaderos positivos de ambas métricas y representado en la siguiente fórmula:

$$F1 - Score = 2x \frac{\mathrm{Pr}\,ecision \times Recall}{\mathrm{Pr}\,ecision + Recall}$$

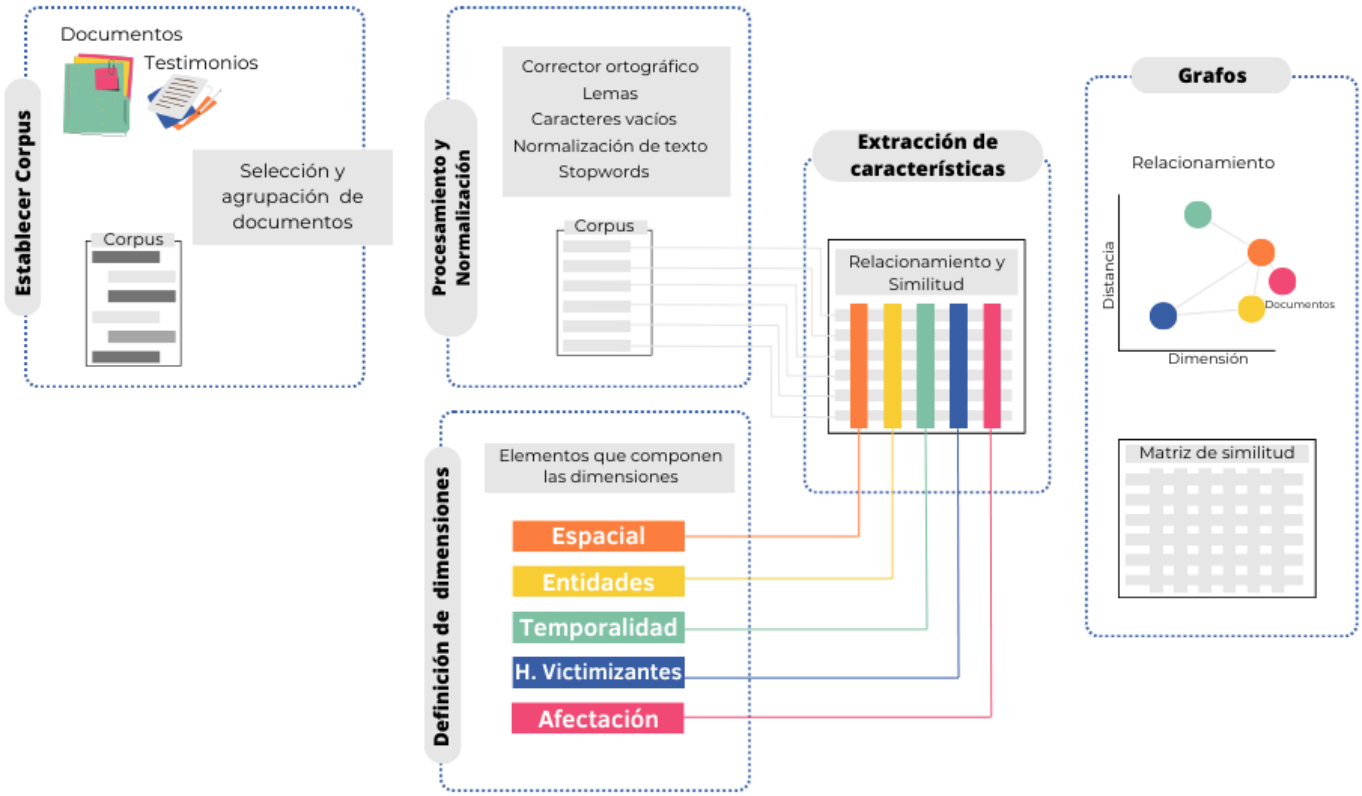
5 Metodología

Este proyecto se desarrolla a través de cinco fases, las cuales van desde la adquisición de una base de datos de testimonios, la cual se consolida en un *corpus* testimonial y corresponde a la fase inicial; en la segunda fase se realiza el tratamiento previo a los datos a fin de estandarizarlos, llamada fase de preprocesamiento y limpieza; la fase tres corresponde a la identificación de las dimensiones, en las que se registran y condensan las relaciones entre documentos. El tratamiento o manipulación de los datos se realiza en la cuarta fase por medio de la aplicación de código y algoritmos en lenguaje de programación Python. El cierre de las actividades se presenta en la quinta fase con la presentación de resultados, tanto en forma matricial como de *grafos*.

En la **Figura 2** se ilustra la aplicación del método para la identificación de las relaciones entre los textos del *corpus*.

Figura 2

Esquema general de desarrollo del proyecto. Metodología para el relacionamiento automático de testimonios



5.1 Fase 1: consolidación del *corpus* testimonial

Se considera relevante resaltar la definición de las víctimas acuñada por la *Ley de Justicia y Paz,* Artículo 5 de la Ley 975, señalando que la víctima del conflicto es un sujeto violentado y con derecho a ser reparado (Colombia. Congreso de la República, 2005). Por su parte, se describen 11 categorías de violencia documentadas en el marco del conflicto armado, acontecidas entre 1958 y 2020, a saber: acciones bélicas, asesinato selectivo, ataques a poblados, atentado terrorista, daño a bienes o civiles, masacre, secuestro, desaparición forzada, reclutamiento, violencia sexual y minas antipersonal y munición sin explotar (Colombia. Centro Nacional de Memoria Histórica. Observatorio de Memoria y Conflicto, 2020).

Asimismo, en el marco de procesos de resiliencia, perdón y búsqueda de la verdad, las víctimas han narrado voluntariamente sus vivencias acontecidas en medio de la guerra, relatos que configuran el objeto de interés de esta investigación: los testimonios, los cuales han sido recogidos de diversas formas, como por ejemplo narraciones orales, entrevistas, documentales, entre otros, y condensados de igual forma en variedad de formatos tales como escritos, grabaciones sonoras, videográficas, imágenes y otros.

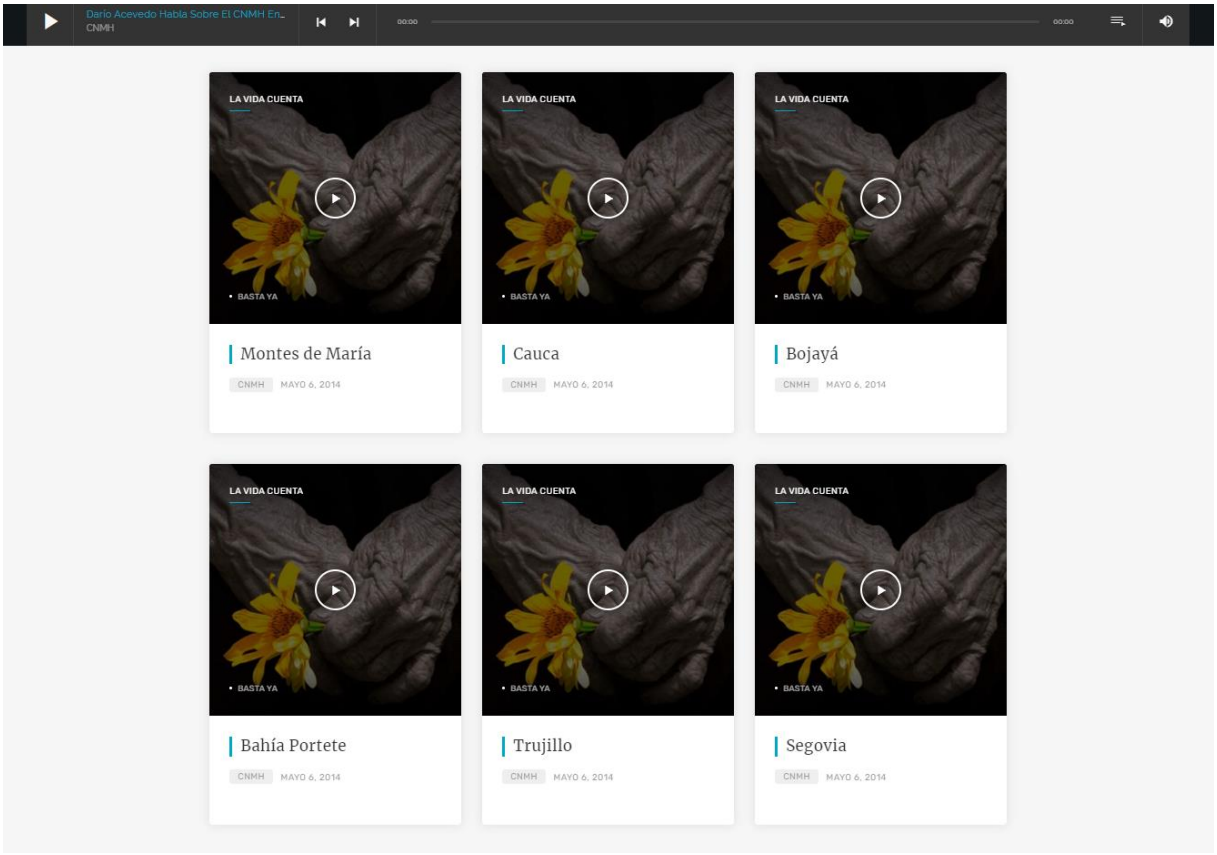
Disponer de un amplio registro de narraciones que dan cuenta de los flagelos de la guerra entre los actores principales (víctimas y victimarios), representa un desafío importante para investigadores e interesados, al tener la posibilidad de identificar relatos similares o de perspectivas distintas de un mismo hecho, pues se convierte en un insumo significativo que aporta al esclarecimiento de la verdad, otorgar el perdón y evitar la repetición. Puesto que estas narraciones están representadas en multiplicidad de formas, se requiere recoger y unificar los testimonios en formato de texto a fin de que procesos automatizados de análisis de estos archivos puedan encontrar elementos comunes ocultos entre las narraciones de las víctimas y que son considerados de interés en tareas complejas como las referenciadas en la Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición (CEV), el Centro Nacional de Memoria Histórica (CNMH) y la Unidad de Búsqueda de Personas dadas por Desaparecidas (UBPD).

Es importante señalar que al inicio de la investigación en el año 2018, el Proceso de Paz en Colombia entre el Estado y la guerrilla de las FARC-EP se encontraba en fase de negociación, por lo que a pesar de existir un gran y disperso número de testimonios de víctimas, aún no se contaban con bases de datos públicas disponibles para su análisis, pues estos eran considerados de carácter reservado, al tratarse de información sensible sin aplicar procesos de anonimización, autorización de licencias o de uso explícito. Como solución a esta necesidad, se identificó que el micrositio web del Centro Nacional de Memoria Histórica, en la sección de Pódcast, tiene disponible una serie de grabaciones con narraciones de víctimas del conflicto armado (Colombia. Centro Nacional de Memoria Histórica, 2020), las cuales fueron descargadas y transcritas a texto a través de un proceso automatizado, para posteriormente efectuar un proceso de revisión manual de revisión de los documentos.

A fin de contar con documentos para la aplicación del método propuesto en este trabajo, se seleccionó la serie *La vida cuenta*, que contiene 6 archivos de audio: Montes de María, Cauca, Bojayá, Bahía Portete, Trujillo y Segovia, publicados en mayo de 2014, con una extensión promedio de 10 minutos por archivo. La selección de estos ficheros de audio se realizó considerando que contienen narraciones de hechos violentos del conflicto armado, descritos por las víctimas a partir de sus vivencias en distintas regiones del país. La estructura del audio comprende una narración en formato libre, acompañada por un narrador, es decir, las víctimas en sus voces cuentan su historia libremente mientras en intermedios y a otra voz, el narrador hace una descripción para entregar mayores detalles y contexto a lo comunicado.

Figura 3

Sección Pódcast del micrositio del Centro Nacional de Memoria Histórica



Fuente. (Colombia. Centro Nacional de Memoria Histórica, 2019).

El proceso de transcripción se realizó de manera automática a través del software genérico online de conversión de voz a texto (Speechnotes, 2024), el cual, al reproducir el audio e iniciar la grabación, la transcripción se presenta en pantalla, por lo que posteriormente se copia el contenido y se lleva como documento de texto. A pesar de tener un alto porcentaje de efectividad, un intérprete comparó el audio con el texto generado en una escucha y lectura simultánea, con el propósito de asegurar la calidad del escrito y corregir errores en la sintáctica y semántica omitidos en esta primera actividad. Como resultado final se obtuvo un número de archivos de texto plano similar al de los testimonios (seis), nombrando el fichero con el mismo título del pódcast, con extensión TXT. El archivo se estructura en texto plano sin asociar algún tipo de formato especial, es decir, sin tablas, alineaciones, colores, etc. La distribución del contenido se representa por párrafos, marcando un salto de línea en cada idea o cambio de voz o emisor (persona que habla en la grabación). Los audios transcritos se encuentran incluidos en los anexos del actual trabajo.

Conforme la investigación avanzó en el tiempo, se rastrearon otras fuentes de información disponibles, tales como las transcripciones de entrevistas anonimizadas resultado del proceso de escucha de la CEV, las cuales fueron publicadas por este organismo (Colombia. Comisión de la Verdad, 2023), acompañando la entrega del Informe final de la Comisión de la Verdad (Colombia. Comisión de la Verdad, 2022) y recuperadas desde el micrositio (Colombia. Comisión de la Verdad, 2024a) en formatos PDF y JSON.

El resultado de esta actividad es la consolidación de un expediente con 2500 archivos de texto que contienen testimonios de víctimas del conflicto armado en Colombia, con lo que se consolida el objetivo específico 2 de esta investigación.

5.2 Fase 2: preprocesamiento y limpieza

A partir de esta fase se da inicio al desarrollo de código en lenguaje de programación Python, marco de trabajo seleccionado al ser considerada una herramienta de fácil uso y propicia para el manejo y representación de datos. A lo largo del trabajo se describen las funciones, los paquetes y las librerías requeridas para ejecutar las acciones. Asimismo, el código se encuentra disponible para su consulta en repositorio GitHub[1].

5.2.1 Definición del entorno y marco de desarrollo

La programación y ejecución de los algoritmos de esta investigación fueron realizados en un entorno virtual de Python, utilizando la plataforma Jupyter Notebook, usando la distribución de Anaconda, en una máquina Lenovo IdeaPad 330s, procesador Intel® Core i5-8250U, 1.60 GHz de CPU, 8 núcleos, 8.0 GB de memoria RAM, 256 GB de disco en estado sólido y sistema operativo Linux, distribución Ubuntu 22.04.3 LTS a 64 bits.

[1] https://github.com/fbaenahe/relacionamiento

Tabla 1
Especificaciones de máquina en la que se desarrolló y ejecutó el algoritmo de relacionamiento de testimonios

Modelo de hardware	Lenovo IdeaPad 330s
Memoria RAM	8.0 GB
Procesador	Intel® Core i5-8250U CPU @ 1.60 GHz x 8
Gráficos	Mesa Intel ® UHD Graphics 620 (KBL GT2)
Capacidad del disco	240,1 GB
Nombre sistema operativo	Ubuntu 22.04.3 LTS
Tipo de sistema operativo	64 bits

5.2.2 Instalación e importación de dependencias y librerías

Para el procesamiento de texto, como lo representa el objetivo de la presente investigación, se requiere de herramientas que estén entrenadas para entender y gestionar la estructura del lenguaje, y en este caso particular considerar el idioma español, por lo que, entre un conjunto de herramientas disponibles para *Python*, se selecciona *Spacy*, por tener, entre otras características, la de ser una de las librerías más utilizadas en el trabajo con el tratamiento de lenguaje natural, pues ha demostrado una alta precisión en funciones de etiquetado, identificación de partes del texto y tokenización; además de contar con modelos preentrenados en múltiples idiomas, entre los que se incluye el español (Al Omran & Treude, 2017). Para el uso de la librería se requiere de descarga, instalación e importación en el entorno de ejecución. Asimismo, la asignación de un modelo entrenado para el proceso de datos textuales, por lo que se carga de entre las estrategias, el modelo más ligero: *es_core_news_sm*, pues este cuenta con las características para el idioma español y requiere de menor demanda de recursos de máquina. De este mismo modelo también se carga la librería *stop_words* para la gestión de palabras comunes o con poca representación semántica desde el punto de vista del análisis del contenido de los documentos, tales como preposiciones y artículos;

así mismo, también es importante resaltar que esta librería también contiene elementos gramaticales tales como verbos y adjetivos.

Otras librerías de procesamiento de lenguaje natural también fueron importadas a este proyecto, tales como S*klearn*, la cual hace parte de la biblioteca de código abierto Scikit-Learn[2], que incluye algoritmos y funciones de aprendizaje supervisado y no supervisado, como el modelo *TD-IDF*, y cuyo uso principal en este trabajo es la del proceso de identificación de similitud en los documentos, procedimiento que se describe con detalle más adelante; adicionalmente, S*cipy,*[3] utilizada para la identificación de grupos o clústeres jerárquicos y su ilustración a través de grafos de dendrogramas, los cuales representan las relaciones existentes entre documentos enlazados por líneas y que se describe en futuras páginas. Asimismo, se cargan aplicaciones de uso más general, entre ellas *Pandas* y *Numpy*, por tratarse de librerías especializadas en la gestión de aplicaciones matemáticas, tablas, matrices y vectores; *matplotlib* para la representación gráfica de los datos y las librerías *os* e *io* para la gestión de los archivos de texto.

[2] https://scikit-learn.org/stable/getting_started.html
[3] https://docs.scipy.org/doc/scipy/index.html

Figura 4
Código de instalación y declaración de librerías necesarias en el procesamiento de lenguaje natural

Instala dependencias

```
pip install -U spacy
```

```
!python -m spacy download es_core_news_sm
```

Importa dependencias

```
import spacy
# Modelo entrenado para idioma español
spc = spacy.load('es_core_news_sm')
# Importa Stop Words
from spacy.lang.es.stop_words import STOP_WORDS
import pandas as pd
pd.options.display.max_colwidth = 1000
%matplotlib inline
import matplotlib.pyplot as plt
import os
from os import listdir
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from scipy.cluster.hierarchy import dendrogram, linkage
from io import open
```

5.2.3 Carga del corpus

Si bien en este punto se cuenta con un *corpus* de testimonios de víctimas del conflicto armado en Colombia, este está representado en unidades testimoniales por archivo individual, es decir, cada archivo de texto contiene la narración de una víctima, por lo que se desarrolla una función capaz de identificar los nombres de los archivos contenidos en una ruta de almacenamiento, cuya extensión corresponda al tipo TXT, para posteriormente leerlo y llevar el contenido a una variable de tipo diccionario.

Figura 5

Código para la carga de documentos de texto que consolida el corpus

Carga de documentos para establecer el corpus

```
# Ubicación de carpeta con testimonios
path = os.getcwd()+"/testimonios/"
# Crear lista vacía para los testimonios
lista_archivos=[]
# Ciclo for para identificar los archivos que tengan punto en su nombre
# el path es la carpeta testimonios
for archivos in listdir(path):
    # Si se encuentran archivos con extensión .txt
    if (".txt" in archivos):
        # Cárguelos a la lista que inició vacía
        lista_archivos.append(archivos)
```

```
# Carga corpus a variable
# Crear lista vacía
corpus=[]
# Ciclo for para identificar cada archivo que está en la variable (de tipo lista) lista_testimonios
for testimonio in lista_archivos:
    # Abra cada testimonio en formato lectura
    testim = open(path+testimonio,"r")
    # Cargue el texto completo
    lee_texto = testim.read()
    # Cierre el archivo que se abrió
    testim.close()
    # Lleve el texto completo (testimonio a la lista de corpus)
    corpus.append(lee_texto)
```

Finalmente se presenta una tabla con los nombres de los archivos, la cantidad de palabras que tiene cada uno y el testimonio.

5.2.4 Funciones de preprocesamiento

Corrector ortográfico: aunque para este trabajo se parte de la premisa de que los testimonios cuentan con una estructura semántica y sintáctica correcta, es importante considerar el uso de un modelo de corrección de ortografía, a fin de modificar palabras con escritura dudosa, principalmente para nuevos documentos en los que no se ha realizado una revisión gramatical. En la actualidad existe una extensa variedad de alternativas en correctores de ortografía, entre ellos PySpellChecker, TextBlob, Aspell, OrTools, Language-Tool-Python, solo por mencionar algunos.

La inclusión de una función de preprocesamiento en el código fue adaptada de la de un corrector ortográfico que ejecuta un algoritmo con la distancia de Levenshtein (Flores, 2019; Maxwell, 2022).

Figura 6
Función en Python para aplicar un corrector ortográfico

```
# Corrector ortográfico
# Inicialmente debe importar el módulo corrector desde el archivo
from ortografia.corrector_ortografico import correction
def corrct(texto):
    cp_funct_cror = []
    for k in texto:
        # Tokeniza palabras
        tk = spc(str(k))
        # realiza la correción ortográfica
        tkn = [correction(str(o)) for o in tk]
        # Crea nuevamente la oración
        tkn = ' '.join(list(tkn))
        cp_funct_cror.append(tkn)
    return cp_funct_cror
```

Lematizador: puesto que el método a implementar en esta investigación considera de gran importancia la frecuencia de los tokens, elementos con semántica similar pero escritos de forma diferente, entre algunos motivos identificados, pueden ser consideradas como diferentes debido a la conjugación verbal, y por tanto tal decisión puede apuntar a un sesgo (a favor o en contra) de una relación entre documentos. Con el propósito de relacionar palabras similares, se incluye el uso de un lematizador, el cual modifica una palabra en el lema y de esta manera tratar de encontrar mayores relaciones entre los testimonios.

Figura 7
Función en Python para lematizar palabras

```
# Lematizar
def lmt(texto):
    cp_funct_lmt = []
    for k in texto:
        # Tokeniza palabras
        tk = spc(str(k))
        # Agrega elemento a lista si no es caracter especial
        tkn = [o.lemma_ for o in tk]
        # Crea nuevamente la oración
        tkn = ' '.join(list(tkn))
        cp_funct_lmt.append(tkn)
    return cp_funct_lmt
```

Caracteres vacíos: los espacios prolongados continuos generan ruido en el texto escrito, por lo que se construye una función que modifica el documento eliminado más de un espacio vacío contiguo.

Figura 8
Función en Python para eliminar caracteres de varios espacios simultáneos

```
# Elimina espacios en blanco
# Esta función tiene una alternativa en la función que engloba el preprocesamiento
# No es necesario cargar en funcion global
def whitespaces(texto):
    cp = [e.strip() for e in texto]
    return cp
```

Normalización de texto: puesto que el método de similitud de documentos implementado considera como diferentes dos elementos (palabras o caracteres) con igual semántica, pero con gramática distinta, se procede a normalizarlo en caracteres en minúscula. De igual forma, este mismo bloque de código elimina los caracteres considerados como especiales, entre ellos el asterisco (*), signos de interrogación (¿)(?), exclamación (¡)(!), entre otros. En idiomas diferentes al español, los caracteres tildados o eñes (ñ) son considerados especiales. Sin embargo, como en el actual proyecto se consideran los testimonios en el idioma español, tales caracteres hacen parte de la gramática aceptada y se conservan.

Figura 9

Función en Python para la normalización de texto en minúsculas y eliminación de caracteres especiales

```
# ELiminar caracteres especiales y pasar a minúsculas
def crhsp_and_case(texto):
    cp_funct_sp_crh_case = []
    for k in texto:
        # Tokeniza palabras
        tk = spc(str(k))
        # Agrega elemento a lista si no es caracter especial
        tkn = [o.orth_.lower() for o in tk if not o.is_punct]
        # Crea nuevamente la oración
        tkn = ' '.join(list(tkn))
        cp_funct_sp_crh_case.append(tkn)
    return cp_funct_sp_crh_case
```

Stopwords: todos los idiomas tienen presentes palabras comunes que ayudan a conectar ideas o frases y agregan estilo y comodidad a la interpretación; sin embargo, si estas se suprimen o ignoran en el análisis del texto, no modifican (por lo menos, en mayor medida) el sentido del discurso, además de facilitar el análisis automático. Por tanto, con la intención de eliminar palabras que no aporten mayor información a los testimonios y agilizar el proceso de identificación de similitud, se implementa una función que elimina las palabras de esta categoría, mismas que están identificadas en el modelo importado de *Spacy*, y al que también se pueden adicionar nuevas o retirar a voluntad.

Figura 10
Función en Python para eliminación de stopwords

```
# ELiminar Stopwords
def stp(texto):
    cp_funct_stp = []
    for k in texto:
        # Tokeniza oraciones
        k = spc(str(k))
        tkn =[]
        # Tokeniza palabras
        for tkw in k:
            tkw = str(tkw)
        # Agrega elemento a lista si no es caracter especial
            if tkw not in STOP_WORDS:
                tkw = str(tkw)
                tkn.append(tkw)
        # Crea nuevamente la oración
        tkn = ' '.join(list(tkn))
        cp_funct_stp.append(tkn)
    return cp_funct_stp
```

Unificando el preprocesador: si bien las funciones hasta aquí mencionadas pueden ser ejecutadas de manera individual, esto representa una tarea compleja y dispendiosa para procesar cada documento, aumentando tal complejidad entre mayor número de documentos se sume para usar. Por lo anterior, se desarrolla una nueva función encargada de consolidar las anteriores, tomando como entrada la variable en la que se cargaron todos los testimonios del *corpus*, aplicando las funciones de preprocesamiento y limpieza a cada elemento de la lista, y devolviendo un archivo más liviano y con las características establecidas para encontrar las relaciones y similitudes.

Figura 11

Función en Python que consolida las funciones de preprocesamiento y limpieza de texto

```python
# Función que se encarga de realizar todo el prepocesamiento
def total (texty):
    ## Normalización
    # Ortografía
    texty = corrct(texty)
    # Lemmas
    texty = lmt(texty)
    ## Limpieza
    # Eliminar espacios en blanco 1
    texty = [e.strip() for e in texty]
    # Caracteres especiales y conversión a minúsculas
    texty = crhsp_and_case(texty)
    # Eliminar Stopwords
    texty = stp(texty)
    # Eliminar espacios en blanco 2
    texty = [e.strip() for e in texty]
    return texty
```

Es preciso señalar que aplicar estrictas reglas de preprocesamiento al *corpus* también puede representar una dificultad, pues al tratarse de funciones entrenadas con textos particulares está presente la probabilidad de errar en coincidencias; por ejemplo, sería posible que una palabra correcta se transforme en otra ajena al contexto, lo que cambiaría el sentido al documento y por tanto alterando las relaciones. Este falso positivo es común en la aplicación de lexemas y correctores de ortografía en palabras asociadas a los nombres propios.

De igual forma, conservar el contexto se convierte en un reto incluso para una palabra bien escrita desde el comienzo, pues relacionado con la pragmática en el lenguaje, esta puede representar una idea diferente.

5.3 Fase 3: definición de dimensiones

Las dimensiones hacen referencia a taxonomías usadas para categorizar palabras comunes, las cuales se emplean en esta investigación para encontrar conjunciones o coincidencias entre los testimonios del *corpus*; por tanto, si más de dos testimonios comparten por lo menos una misma

palabra, se puede asegurar que presentan similitud en determinada dimensión. En el código, las dimensiones están dadas por variables de un conjunto o *set* de datos, lo que exige que éstas solo cuenten con elementos únicos eludiendo la duplicidad de palabras. Una vez consolidados los datos en estas variables, por funcionalidad del algoritmo, la variable se transforma en tipo lista.

Figura 12

Función en Python que carga las dimensiones

Carga diccionario con dimensiones

```
# Carga archivo de diccionarios

filename = 'diccionarios/diccionarios.csv'
diccionarios = pd.read_csv(filename, header=0)
nombre_columnas = list(diccionarios.columns.values)
```

```
# Asignar los valores de las dimensiones a variables de tipo lista, eliminado los elementos duplicados
# También se aplica preprocesador

lista_ubicaciones = list(set(total(diccionarios["Ubicaciones"])))
###lista_ubicaciones = list(set(diccionarios["Ubicaciones"]))
lista_hecho_victimizante = list(set(total(diccionarios["Hecho Victimizante"])))
lista_afectacion = list(set(total(diccionarios["Afectación"])))
lista_temporalidad = list(set(total(diccionarios["Temporalidad"])))
lista_entidades = list(set(total(diccionarios["Entidades"])))
```

Por otro lado, aunque al inicializar el algoritmo se carga cada lista con elementos definidos, las variables son dinámicas, pues el investigador tiene la posibilidad de listar, eliminar o gestionar nuevos elementos en cada dimensión. En este trabajo se señalan 5 dimensiones: espacial, de temporalidad, de afectación, de entidad y de hecho victimizante.

- **Dimensión espacial:** corresponde a la clasificación de los elementos del texto que referencian un lugar en el espacio, tales como ciudades, municipios, países, territorios, parajes, veredas, etc. Los elementos de esta dimensión son extraídos de la base de datos del portal de datos abiertos en el que se registran los municipios de Colombia (Colombia. Ministerio de Tecnologías de la Información y las Comunicaciones, 2024). Algunos de los elementos que tiene esta variable son: Pueblorrico, San Roque, Arjona.

- **Dimensión de temporalidad:** en esta dimensión se listan palabras relacionadas con el tiempo, tales como fechas, años, horas, períodos, días, épocas, entre otras.

- **Dimensión de afectación:** esta variable alberga los elementos referentes a las afectaciones de tipo: **a)** político-social, como descuido estatal, extradición, denuncia; **b)** económico-material, como desabastecimiento, pobreza, tugurio; **c)** físicas, como herido; asesinato, violación; **d)** psicológicas, como miedo, susto, depresión.

- **Dimensión de entidad:** comprende nombres propios de personas, grupos, organizaciones, entidades o sujetos a los que se les pueda atribuir una acción, tales como Diego, AUC, Naciones Unidas, mujer.

- **Dimensión de hecho victimizante:** hace mención de los elementos del discurso que señalan una acción que afecta una víctima, entre ellos secuestro, masacre, mutilado.

El proceso para conformar las dimensiones es dado por listados definidos, entre ellos los expresados en las dos primeras dimensiones (espacial y de temporalidad). Por su parte, para las dimensiones entidad, afectación y hecho victimizante, se construyeron adicionando manualmente elementos basados en el trabajo de curaduría de un experto, pues aprovechando el proceso de lectura y revisión de los testimonios de la fase 1, se marcan y clasifican las palabras con significado en estas dimensiones. Es de resaltar que esta actividad demanda gran cantidad y variedad de recursos; sin embargo, se trata de una tarea acumulativa, pues nuevos términos encontrados se van adicionando a las dimensiones cuando el experto lo especifique. El listado de las dimensiones puede ser consultado en los anexos.

Esta sección finaliza con una tabla y una figura que resumen el número de elementos de cada dimensión, así como funciones que le permiten al usuario validar si un elemento está incluido en las dimensiones, y que, en caso de no localizarlo, ofrece la posibilidad de adicionarlo.

Figura 13
Código que presenta un resumen de las dimensiones cargadas

Información parcial del proceso (Dimensiones)

```
# Crea DataFrame de cantidad de elementos por dimensión

infodimensiones = [len(lista_ubicaciones),len(lista_hecho_victimizante),len(lista_afectacion),
                   len(lista_temporalidad),len(lista_entidades)]
dimensiones = ["Ubicaciones","Hecho Victimizante","Afectación","Temporalidad","Entidades"]
dfdimensiones = pd.DataFrame({"Dimensión":dimensiones,"Cantidad":infodimensiones})
dfdimensiones
```

5.4 Fase 4: desarrollo y aplicación del algoritmo

Esta fase del algoritmo es la encargada de gestionar los datos, para lo cual se inicia con el preprocesamiento del *corpus* haciendo uso de las funciones detalladas previamente, y obteniendo como resultado una variable con un nuevo *corpus* normalizado, además presentando una tabla y una figura con el resumen comparativo de los datos iniciales.

Figura 14
Código en Python que realiza el preprocesamiento y limpieza del corpus

```
[ ]: # Preprocesamiento de corpus
corpuslimpio = total(corpus)
```

```
[ ]: # Dataframe descriptivo
tamanocorpuslimpio =[len(amp) for amp in corpuslimpio]
tottamanocorpuslimpio = sum(tamanocorpuslimpio)

tot_dif_corpus = np.array(tamanocorpus) - np.array(tamanocorpuslimpio)
porc_dif_corpus = np.round((tot_dif_corpus/np.array(tamanocorpus))*100,decimals=2)
```

```
[ ]: print("""
    Resumen de Corpus
        Se cargaron """ + str(len(lista_archivos)) + """ archivos de texto de la carpeta testimonios
        La carga inicial contiene """ + str(tottamanocorpus) + """ palabras
        El corpus procesado contiene """ + str(tottamanocorpuslimpio) + """ palabras
        Correspondiente a una diferencia de {:.2f} %
        """.format((tottamanocorpuslimpio/tottamanocorpus)*100))
```

Figura 15

Código que grafica el resumen del preprocesamiento y limpieza inicial del corpus

```
# Grafica de dimensión de corpus por testimonio
# Si el número de testimonios supera los 15 pasa de barras a líneas

plt.figure(figsize=(len(lista_archivos)+3, 4))
if len(lista_archivos) > 15:
    plt.plot(lista_archivos,tamanocorpus, color = "mediumaquamarine", label = "Sin procesar")
    plt.plot(lista_archivos,tamanocorpuslimpio, color = "darkslategray", label = "Procesado")
    plt.legend(fontsize=10, shadow=True, ncol=2, loc=6)
else:
    plt.bar(lista_archivos,tamanocorpus, color = "mediumaquamarine", label = "Sin procesar")
    plt.bar(lista_archivos,tamanocorpuslimpio, color = "darkslategray", label = "Procesado")
    plt.legend(fontsize=10, shadow=True, ncol=2, loc=3)
plt.suptitle("Tamaño del Corpus (palabras por documento)")
for i,j in zip(lista_archivos,tamanocorpus):
    plt.annotate(j,xy=(i,j+50),color="darkslategray")
for i,j in zip(lista_archivos,tamanocorpuslimpio):
    plt.annotate(j,xy=(i,j+50),color="darkslategray")
plt.show()
```

Posteriormente, se procede a identificar los elementos coincidentes de los testimonios por cada dimensión, por lo que una función recorre y compara cada palabra de los testimonios en todas las dimensiones, generando una nueva variable de tipo diccionario con los elementos encontrados, es decir, que, si al validar una palabra en una dimensión encuentra coincidencia, esta es llevada a una nueva lista. El resultado de este código es la consolidación de un *dataframe* de la librería *Pandas* el cual se compone del cruce de los elementos de cada dimensión por todos los testimonios, es decir, una matriz con las dimensiones en sus columnas y los archivos en las filas, el cual es fuente para establecer las relaciones.

Por su parte, ya que el actual *dataframe* cuenta con los elementos de cada dimensión en su frecuencia original, es decir, que si una palabra fue recuperada más de una vez, esta se repite en la variable el mismo número de veces, y puesto que el método aplicado a este trabajo determina que existe una relación entre dos documentos si por lo menos un elemento de una dimensión se encuentra en ambos, a fin de economizar recursos y hacer el proceso más ágil, se genera un nuevo *dataframe* con los datos únicos, para posteriormente recorrer de nuevo cada elemento en todas las dimensiones para todos los documentos y determinar relaciones.

Figura 16
Función que unifica datos y cuenta frecuencias en un dataframe

```
# Función para unificar valores y contar frecuencias
# devuelve dos variables
def unificarfrecuencias(objeto):
    Val_Unic = []
    Cont_Unic = []
    for sublista in objeto:
        Val_Unic_tmp = []
        Cont_Val_tmp = []
        for item in sublista:
            if item not in Val_Unic_tmp:
                Val_Unic_tmp.append(item)
                Cont_Val_tmp.append(sublista.count(item))
        Val_Unic.append(Val_Unic_tmp)
        Cont_Unic.append(Cont_Val_tmp)
    return Val_Unic, Cont_Unic
```

5.4.1 Relaciones

Para establecer las relaciones, se asignan niveles a los datos, marcando el nivel alto (o de tipo 1) a los elementos generales, es decir, a los testimonios; y un nivel bajo (o de tipo 3) al dato más específico, en este caso a los elementos identificados en las dimensiones, las cuales se encuentran en un nivel intermedio (o de tipo 2). Por ejemplo, elementos identificados como Bojaya.txt, Segovia.txt y Montes_de_maria.txt hacen referencia a los testimonios, es decir, a los documentos, por lo que se convierten en elementos de tipo 1. Por su parte, las palabras Afectación, Entidades, Ubicación, Hecho victimizante y Temporalidad son los elementos de tipo 2, ya que nombran las cinco dimensiones que se identifican en el algoritmo. Finalmente, palabras como Masacre o Carmen, son ejemplos de algunas que se encuentran contenidas en las dimensiones Hecho victimizante y Entidades, y por tanto inmersas en elementos de tipo 3.

Así pues, si un elemento de una dimensión, en un documento, tiene coincidencia en otro documento se establece una relación de tipo 3 entre los elementos y, por consiguiente, también una relación de tipo 2 entre las dimensiones comunes de cada documento; así como una relación de tipo 1 entre los documentos. En un sentido más general, dado que dos documentos comparten por lo menos un elemento en común y por lo menos una dimensión, es posible afirmar que tales documentos están relacionados. Por ejemplo, partiendo del supuesto de que los documentos Bojaya.txt y Segovia.txt tienen inmersa la palabra "Masacre", es posible asegurar que éstos (los documentos) tienen una relación de tipo 1, pues su elemento coincidente (la palabra) es un elemento que se encuentra en el nivel más bajo.

Para marcar las relaciones se establece un código único a cada documento, dimensión y elemento, compuesto por una letra mayúscula y un valor numérico. Las letras corresponden a D para el nivel 1, M para identificar las dimensiones y E para asociar los elementos o el nivel 3. Por su parte, el valor numérico corresponde al lugar que ocupa un elemento en una lista ordenada alfabéticamente para su nivel, por ejemplo, los códigos D1, D2, M3 y E4 identifican los documentos (o testimonios) 1 y 2; la dimensión 3 y el elemento 4.

La codificación tiene como característica heredar el código del nivel anterior, lo que quiere decir que un código de nivel 2 se compone por dos letras y dos valores numéricos, conservando su estructura, es decir, *Letra - Valor y Letra – Valor*; y para un código de nivel 3 se compone de tres letras y tres valores numéricos, conservando igualmente su estructura, de la forma *Letra – Valor, Letra - Valor y Letra – Valor*; por lo tanto, el código D3M2E1, corresponde al elemento número 1, del documento 3, en la dimensión 2.

Figura 17

Función en Python que codifica los elementos de un testimonio

```
# Función que obtiene las interrelaciones entre los términos de todos los documentos de la misma dimensión
# Recibe como parámetro el DF del listado de términos discriminados

def relintraitem(listado):
    EntreItems = []
    EntreDocs = []
    for M, dime in enumerate(listado):
        for d, doc in enumerate(listado.iloc[:,M]):
            D = d
            for i, item in enumerate(doc):
                v = D+1
                for j, sublist in enumerate(listado.iloc[v:,M]):
                    for k, subitem in enumerate(sublist):
                        if item == subitem:
                            a = "D"+str(D)+"M"+str(M)+"E"+str(i)
                            b = "D"+str(D+j+1)+"M"+str(M)+"E"+str(k)
                            c = "D"+str(D)
                            e = "D"+str(D+j+1)
                            EntreItems.append((a,b))
                            EntreDocs.append((c,e))
    #Relaciones inter documentos deshabilitadas
    EntreDocs = list(set(EntreDocs)) # Relación entre documentos
    EntreItems = EntreItems + EntreDocs # Relación entre documentos
    return EntreItems
```

Las relaciones se recogen en una lista de tuplas, para lo cual cada una de ellas se compone de dos elementos, pues cada uno corresponde a código de niveles similares o diferentes. Por ejemplo, la tupla (D1M2, D1M2E1) indica que hay relación entre un elemento y su propia dimensión, es decir, el elemento 1, de la dimensión 2, del documento 1, y la misma dimensión del documento referenciado.

5.4.2 Similitud

Dado que a este punto ya fue posible establecer relaciones, es importante conocer qué tan cercanas o distantes están unas de otras, para lo cual se requiere la consolidación de un *corpus* solo con los elementos que coinciden en las dimensiones. A fin de tener control sobre los documentos, se establece nuevamente un comparativo entre el número de palabras o elementos del *corpus* inicial, el preprocesado y el nuevo resultado del mencionado en este párrafo.

A este *corpus* se le aplica la técnica TF-IDF (ya referenciada en el marco teórico), con el objetivo de comparar e identificar las similitudes de cada texto entre todos los testimonios. A esta técnica se le asigna valor 1 a 3 en el parámetro *ngram_range*, el cual consiste en analizar en cada testimonio un elemento con el siguiente y el siguiente término colindante, es decir, se considera un término de manera independiente, el mismo término con el siguiente y, finalmente, sumado a los dos anteriores, el término siguiente. El resultado de esta técnica es un *dataframe* con un número de filas correspondiente a los documentos analizados, y cuyo cruce en las columnas corresponde a las coincidencias entre términos individuales, pares y de tres elementos o términos consecutivos de cada documento.

Figura 18
Algoritmo de técnica TD-IDF

```
identificador = TfidfVectorizer(ngram_range=(1, 3), min_df=0., max_df=1., norm='l2',
                    use_idf=True, smooth_idf=True)
mapa_elementos = identificador.fit_transform(testim)
mapa_elementos = mapa_elementos.toarray()

etiq = identificador.get_feature_names_out()
pd.DataFrame(np.round(mapa_elementos, 2), columns=etiq)
```

A fin de identificar qué tan similar es un documento de otro, se emplea la medida de similitud de coseno, obteniendo como resultado un *dataframe* de *n x n*, siendo *n* un número de columnas igual al número de filas, las cuales, a su vez, corresponden al mismo número de documentos o testimonios analizados, pues se trata de la comparación de similitud de un testimonio frente a los demás y entregando como resultado la distancia, es decir, un valor que representa qué tan idéntico (cercano a 1) o diferente (alejado de 1) puede ser uno de otro. Por tanto, su diagonal siempre será igual a uno (1), es decir, exactamente igual, pues se trata de la comparación del mismo documento.

Figura 19
Distancia de Coseno. Función que calcula la similitud entre documentos

```
# Se mide la distancia con la Similaridad de Coseno
similaridad = cosine_similarity(mapa_elementos)
df_similaridad = pd.DataFrame(similaridad)
df_similaridad
```

5.5 Fase 5: presentación de resultados

Con la intención de expresar en grafos las relaciones identificadas en las secciones anteriores, se proponen dos estrategias que permiten su interpretación:

La primera muestra la similitud de los testimonios, estableciendo grupos a través de un dendograma, utilizando la función *linkage* y considerando la medida de distancia de Ward. Para su visualización se grafica utilizando la librería *myplotlib*.

Figura 20
Análisis de similitud a grupos o clústeres de documentos

```
# Análisis de similaridad a partir de matriz de dendograma
tabla_dendograma = linkage(similaridad, 'ward')
pd.DataFrame(tabla_dendograma, columns=['Enlace A', 'Enlace B',
                         'Distancia', 'Tamaño de cluster'], dtype='object')
```

Figura 21
Código que grafica en un dendograma jerárquico los clústeres de documentos

```
# Gráfico en dendograma
plt.figure(figsize=(15, 5))
plt.title('Dendograma de Clustering Jerárquico',fontsize=14)
plt.xlabel('Testimonios', fontsize=13)
plt.ylabel('Distancia', fontsize=13)
dendrogram(tabla_dendograma)
plt.axhline(y=1, c='red', ls='--', lw=0.5)
```

La segunda estrategia presenta las relaciones que se expresan a través de la librería *Pyvis*[4], un visualizador de grafos de redes sociales interactivo, para lo cual los elementos de cada nivel previamente identificados toman el parámetro de nodos, en tanto que las relaciones conservadas en las tuplas aplican el parámetro de *edges*. Para hacer visualmente más atractivo el grafo, se da forma, peso y color a las relaciones y nodos, para, finalmente, presentarse en un archivo HTML.

Figura 22

Configuración de parámetros en Pyvis que crea grafo de redes para representación de relaciones

```
# Inicializar la red en una instancia
from pyvis.network import Network
#grafo = Network(height='550px', width='100%', bgcolor='#141414', font_color='white')
grafo = Network(height='550px', width='100%', font_color="#10000000")
#grafo.barnes_hut()
```

```
# Asignación de nodos y propiedades
# grafo.add_nodes(identificadores,title=titulos,color=colores) # Sin considerar pesos
grafo.add_nodes(identificadores,size=peso,title=titulos,color=colores)
```

```
# Asignación de relaciones
grafo.add_edges(relacionamiento)
```

```
# Presentación de grafo
grafo.show('nodos.html', notebook=False)
```

[4] https://pyvis.readthedocs.io/en/latest/

6 Resultados

Mientras que la sección anterior se enfoca en detallar la metodología y el proceso, así como describir el código que consolida el *corpus*, dimensiones, preprocesamiento y limpieza, y establecimiento identificación y gráfico de relaciones y similitudes, esta sección presenta la aplicación del método con los datos consolidados, por lo que tanto texto como imágenes ilustran los resultados obtenidos.

Una vez consolidados en una carpeta y en formato TXT los documentos que hacen parte del *corpus* de testimonios, así como estructurado el algoritmo para su análisis, se procede a la ejecución en la máquina y recursos antes mencionados, encontrando los resultados que se mencionan a continuación, siendo importante destacar que la prueba de funcionamiento se realiza con los archivos transcritos de la serie *La vida cuenta*, del pódcast del Centro Nacional de Memoria Histórica, descrito en la parte inicial del capítulo anterior.

A fin de establecer un control sobre las relaciones que se determinan entre los documentos, se inserta un texto sintético adicional, es decir, un documento ficticio controlado, el cual contiene palabras intencionalmente coincidentes entre las dimensiones, como lo es el caso de “Municipio”, “Región” y “Segovia” en la dimensión Ubicaciones; "Masacre” en la dimensión "Hecho victimizante”; "Muerto” y “Violencia” en la dimensión Afectación; “Muerto” y “Profe” en la dimensión Entidades; y sin ningún elemento en la dimensión Temporalidad. El resumen del *dataframe* del *corpus* inicial da cuenta de siete archivos: seis correspondientes al texto (transcritos), más el sintético controlado.

Figura 23

Presentación en dataframe del corpus inicial de testimonios

	Archivo	Palabras	Testimonio
0	Bojaya.txt	8822	Mi nombre es Rosa de la Nieve Mosquera Cuello, lo que soy se lo debo a Bojayá y me desvivo por Bojayá \nRosa nació en Necoclí en la región de Urabá, a los 7 años se mudó con su familia a Bojayá un municipio en el departamento del Chocó, al noroeste de Colombia, en la región del Pacífico. Rosa es sobreviviente de lo que se conoce como la masacre de Bojayá, ocurrida el 2 de mayo del 2002, ese día murieron 79 civiles de los cuales 48 eran menores de edad. En un combate con los paramilitares, el bloque 58 de las FARC lanzó un cilindro bomba al interior de la iglesia, ese día Rosa estaba con más de 300 personas y sus seis hijos, refugiados en la iglesia. Rosa tienen hoy 48 años, y es la mayor de 6 hermanos, todos fueron criados en Bojayá por su abuela paterna, la relación de Rosa con ella no fue fácil, pues su abuela la maltrataba\nEntonces son, unos de los maltratos que uno aprende a perdonar pero que siempre quedan secuelas, cuando uno se viste y se mira uno dice: esto me lo hizo mi ...
1	Montes_de_maria.txt	9059	Nací en El Carmen de Bolívar, en el barrio El Porvenir ahí vivo aún…\n\nEsta es Soraya Bayuelo, hoy tiene 55 años, y en 1994 junto con comunicadores, maestros y líderes comunitarios del Carmen de Bolívar creó lo que se conoce como el colectivo de comunicaciones “Montes de María”, Soraya nos resume qué es…\n\nNace de las entrañas de la casa de la cultura, es un proyecto con una claridad desde los principios de defensa de los derechos humanos, especialmente el derecho a la comunicación es un proyecto de transformación social, que creíamos que hubiese podido seguir normal, ¿verdad? Si no irrumpe el conflicto armado…\n\nEl conflicto al que se refiere Soraya es a la lucha entre guerrilleros y paramilitares, que empezó con la llegada de la guerrilla a la zona a mediados de los 80’s y la zona es: Montes de María, una región del Caribe interno colombiano ubicado entre los departamentos de Sucre y Bolívar, donde entre 1999 y el 2001 hubo 42 masacres que dejaron 354 civiles asesinados… Tal v...
2	Segovia.txt	7932	Porque con la masacre las víctimas fuimos todos, todo el pueblo de Segovia quedó victimizado con esa masacre, pero ya nosotros quedamos con todas esas secuelas. Era el miedo a la noche, miedo a lo extraño.\nEste es Fernando, mejor conocido como “el profe”, nació en 1962 en Segovia, un municipio del nordeste de Antioquia, un departamento de Colombia. Desde 1982 hasta 1997 Segovia, como la población cercana de Remedios, experimentó una violencia política recurrente contra la población civil, iba dirigida sobre todo hacia los partidos políticos de izquierda: el partido comunista, MOIR, A luchar, Unión patriótica; pero también iba dirigida a los miembros de sindicatos, de comités de Derechos Humanos, juntas cívicas y asociaciones comunitarias. Esta la historia de una de las cuatro masacres más significativas ocurridas en la región y la oiremos a través del testimonio de Fernando. Se trata de la masacre del 11 de noviembre de 1988 donde en una alianza conformada por miembros del ejércit...

Posteriormente, se cargan las funciones de preprocesamiento y se listan las palabras vacías o *stopwords*. Esta función da cuenta del número de palabras cargadas (521), presentando un menú de opciones:

Figura 24

Resumen de palabras vacías o stopwords

```
>>> Se han cargado 521 Palabras vacías o StopWords <<<

    * Sí desea concocer las palabras vacías, presione 1
    * Sí desea consultar si el sistema considera una palabras vacía, presione 2
    * Para contnuar, ingrese cualquier otro número

Su opción: ↑↓ for history. Search history with c-↑/c-↓
```

La primera opción lista las palabras cargadas en la memoria.

Figura 25
Lista de palabras stopwords

```
Su opción:  1
{'aquello', 'da', 'haceis', 'ser', 'consigue', 'buen', 'ademas', 'aquélla', 'o', 'lo', 'debido', 'sin
s', 'esto', 'solo', 'usan', 'sabeis', 'debe', 'largo', 'soy', 'fui', 'ni', 'lado', 'comentó', 'los',
ú', 'aun', 'usas', 'suyos', 'puedo', 'mucho', 'dan', 'entonces', 'tendrá', 'queremos', 'expresó', 'ta
o', 'otro', 'alguna', 'ocho', 'nosotras', 'dijeron', 'algunos', 'hace', 'ahora', 'suyo', 'nueva', 'nu
a', 'éstos', 'va', 'delante', 'apenas', 'nuestro', 'primero', 'dar', 'siendo', 'cuántas', 'dentro', '
'mediante', 'podemos', 'además', 'parte', 'últimas', 'hasta', 'quien', 'adelante', 'unos', 'tienen',
'sobre', 'si', 'próximo', 'solos', 'eres', 'día', 'que', 'tampoco', 'estará', 'buena', 'pasado', 'ésa
ta', 'vuestro', 'propias', 'dónde', 'afirmó', 'parece', 'este', 'manifestó', 'tras', 'por', 'ahi', 'b
o', 'habrá', 'sé', 'realizar', 'vaya', 'podrían', 'segun', 'repente', 'dias', 'habia', 'dio', 'creo',
```

La segunda opción presenta un nuevo cuadro de búsqueda, el cual permite registrar una palabra y validar si existe en el listado.

Figura 26
Módulo de consulta de stopwords

```
>>> Se han cargado 521 Palabras vacías o StopWords <<<

    * Sí desea concocer las palabras vacías, presione 1
    * Sí desea consultar si el sistema considera una palabras vacía, presione 2
    * Para contnuar, ingrese cualquier otro número

 Su opción:  2
Qué palabra desea consultar? ↑↓ for history. Search history with c-↑/c-↓
```

Si la palabra consultada coincide, muestra mensaje de confirmación.

Figura 27
Detalle de la consulta del stopword

```
>>> Se han cargado 521 Palabras vacías o StopWords <<<

    * Sí desea concocer las palabras vacías, presione 1
    * Sí desea consultar si el sistema considera una palabras vacía, presione 2
    * Para contnuar, ingrese cualquier otro número

Su opción:  2
Qué palabra desea consultar?  solo
>> solo << Sí se encuentra!
```

Si la palabra no se encuentra, el código brinda la posibilidad de incluirla.

Figura 28

Módulo de ingreso de stopwords

```
      * Sí desea concocer las palabras vacías, presione 1
      * Sí desea consultar si el sistema considera una palabras vacía, presione 2
      * Para contnuar, ingrese cualquier otro número

Su opción:  2
Qué palabra desea consultar?  fogata
>> fogata << No está en el diccionario!
Desea agrgarla?
Escriba >S< para Sí; cualquier otra opción para No:  S
La palabra fogata fue gregada
```

Se puede observar que al ejecutar nuevamente la misma línea de código se actualiza el número de palabras.

Figura 29

Módulo de consulta de stopwords con elementos actualizados

```
>>> Se han cargado 522 Palabras vacías o StopWords <<<

      * Sí desea concocer las palabras vacías, presione 1
      * Sí desea consultar si el sistema considera una palabras vacía, presione 2
      * Para contnuar, ingrese cualquier otro número

Su opción:  ↑↓ for history. Search history with c-↑/c-↓
```

El siguiente procedimiento corresponde a la carga de dimensiones preestablecidas para, posteriormente, presentarlos en un gráfico y su respectiva tabla. Esta tarea exige el consumo de recursos de máquina, por lo que se identifica como un pico llegando a presentar una breve espera en la presentación de resultados.

Puesto que la dimensión Ubicaciones tiene como fuente de información datos gubernamentales, representa más cantidad de elementos que los demás.

Figura 30
Cantidad de palabras por dimensión

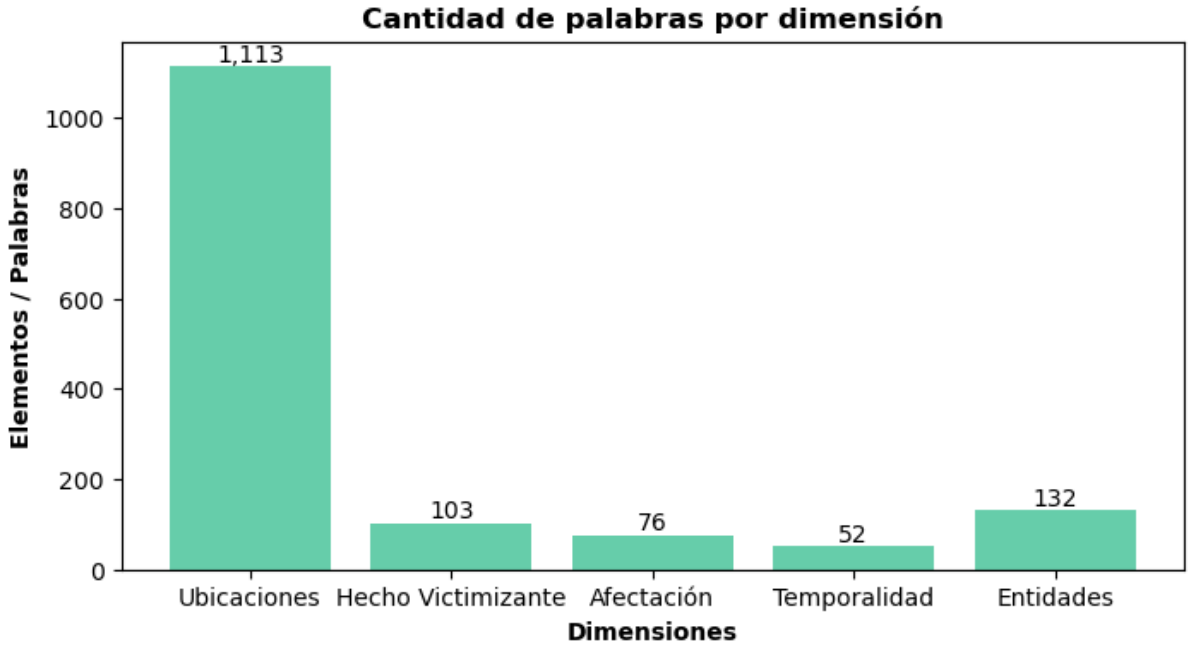


Tabla 2
Cantidad de palabras por dimensión

	Dimensión	Cantidad
0	Ubicaciones	1113
1	Hecho victimizante	103
2	Afectación	76
3	Temporalidad	52
4	Entidades	132

Al igual que con los *stopwords*, también se presenta un menú con alternativas para listar las palabras de cada dimensión y la consulta de una palabra en particular. Por ejemplo, considerando la figura anterior, y al seleccionar la opción 2, se listan los 76 elementos de la dimensión Afectación.

Figura 31
Módulo de consulta de palabras por dimensión: Afectación

```
    * Sí desea ver las palabras de una dimensión, presione el número correspondiente a su índice
    * Sí desea consultar si se ha considerado una palabra en cualquier dimensión, presione 9
    * Para contnuar, introduzca un número que no esté en el listado

Su opción:  2
                       0
0      falta medicamentos
1               dictadura
2        memoria victimas
3                     ira
4   falta equipos médicos
..                    ...
71                    nan
72                 dormía
73                memoria
74               psicosis
```

Por su parte, si se selecciona la opción de consulta (9) buscando la palabra Ejército, presenta el resultado confirmando la dimensión a la que pertenece, en este caso Entidades.

Figura 32
Detalle de la consulta por palabra en dimensiones

```
    * Sí desea ver las palabras de una dimensión, presione el número correspondiente a su índice
    * Sí desea consultar si se ha considerado una palabra en cualquier dimensión, presione 9
    * Para contnuar, introduzca un número que no esté en el listado

Su opción:  9
Qué palabra desea consultar?  Ejercito
>> Ejercito << sí se encuentra en la BD y está en la dimensión:
['Entidades']
```

En caso de no encontrar coincidencia, plantea la posibilidad de agregarla en alguna dimensión.

Figura 33
Módulo de ingreso de palabra a dimensión

```
    * Sí desea ver las palabras de una dimensión, presione el número correspondiente a su índice
    * Sí desea consultar si se ha considerado una palabra en cualquier dimensión, presione 9
    * Para contnuar, introduzca un número que no esté en el listado

Su opción:  9
Qué palabra desea consultar?  maniobra
La palabra no se encuentra en ninguna dimensión, ¿desea agregar >> maniobra << a alguna?
```

A continuación, se aplican las funciones de preprocesamiento al *corpus*, en el que se presenta un resumen general y detallado por dimensión, obteniendo como resultado una reducción de 46,21% de palabras en total.

Figura 34
Resumen del corpus preprocesado

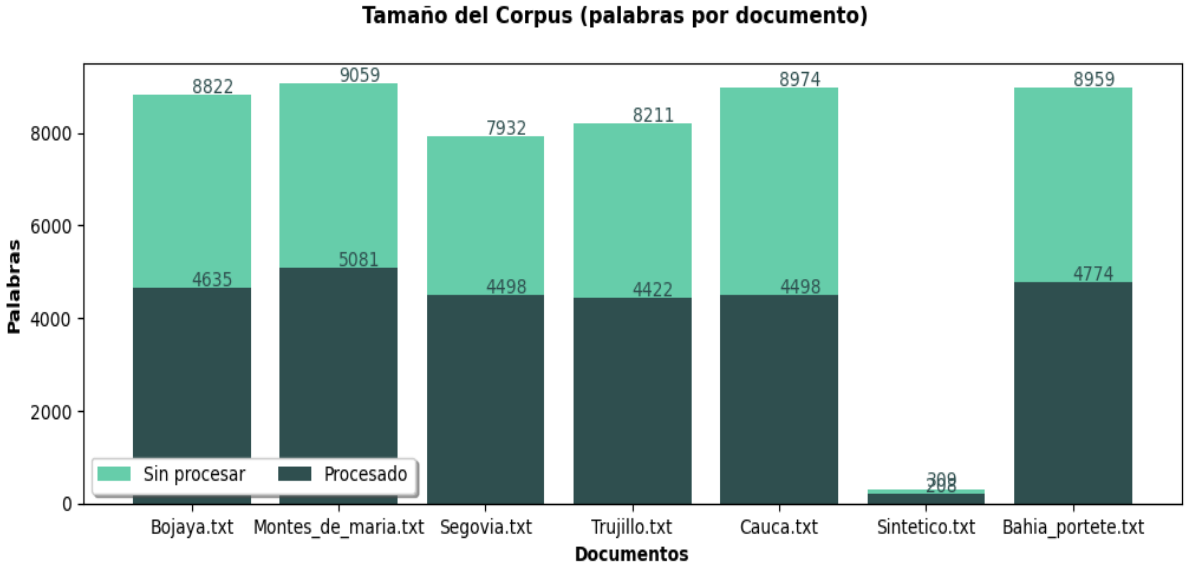
```
Resumen de Corpus
    Se cargaron 7 archivos de texto de la carpeta testimonios.
    La carga inicial contiene 52266 palabras
    El corpus procesado contiene 28116 palabras
    Correspondiente a una diferencia de 53.79 %
```

Tabla 3
Tabla del corpus preprocesado

	Archivo	Testimonio base	Testimonio limpio	Diferencia	% Dif
0	Bojaya.txt	8822	4635	4187	47.46
1	Montes_de_maria.txt	9059	5081	3978	43.91
2	Segovia.txt	7932	4498	3434	43.29
3	Trujillo.txt	8211	4422	3789	46.15
4	Cauca.txt	8974	4498	4476	49.88
5	Sintetico.txt	309	208	101	32.69
6	Bahia_portete.txt	8959	4774	4185	46.71

Figura 35

Comparativo de corpus inicial y preprocesado



Con el *corpus* reducido, se procede a realizar la identificación de los elementos de cada dimensión, inicialmente en listas independientes y posteriormente consolidadas en un nuevo *dataframe*. Se puede observar que los elementos se organizaron por orden alfabético, registrando las palabras según su frecuencia. De igual forma, también se puede evidenciar que todos los testimonios cuentan con elementos en sus dimensiones, excepto los textos *Cauca.txt* y *Sintetico.txt*, los cuales presentan listas vacías en la dimensión Temporalidad. La diferencia entre estos dos textos es que no se encontraron datos, en el primero por simple coincidencia, mientras que en el segundo fue intencional al no reportarse datos para esta dimensión.

Figura 36

Fragmento de dataframe de palabras por dimensión

	documento	Ubicaciones	Hecho Victimizante	Afectación	Temporalidad	Entidades
0	Bojaya.txt	[bellavista, bellavista, bellavista, carretera, carrillo, casa, casa, casa, casa, casa, casa, casa, casa, casa, casa cural, casa cural, casa cural, casa cural, casas, centro salud, centro salud, centro salud, chocó, colegio, colombia, departamento, iglesia, iglesia, iglesia, iglesia, iglesia, iglesia, iglesia, iglesia, iglesia, iglesia, iglesia, mares, medellín, mosquera, municipio, municipio, necoclí, noroeste, orilla, panamericana, región, región, región, región, restaurante, río, universidad, urgencias, vigía fuerte, vigía fuerte, vigía fuerte, vigía fuerte, zona, zona]	[bomba, cilindro, cilindro, cilindro bomba, combate, conflicto, daba, destrozo, explosión, masacre, masacre, masacre, matar, pipeta, refugiados]	[cáncer, esquirlas, herida, herida, herida, heridos, heridos, muerto, muerto, muerto, necesario, salud, salud, salud, secuelas]	[marzo, mayo, mayo, mayo, mayo, mayo, mayo, mayo, mayo, mayo, octubre]	[abuela, abuela, abuela, abuela, aliciente, antún ramos, auria, autoridades, auxiliar, auxiliar, bloque 58, camilo, civiles, comandante, comandante, enfermera, enfermera, enfermera, farc, farc, farc, guerrilla, guerrilla, guerrilla, guerrilla, guerrilla, joven, mamá, mamá, menores, misioneras agustinas, monjas, monjas, muerto, muerto, muerto, niña, paciente, padre, padre, padre, papá, paramilitares, paramilitares, paramilitares, paramilitares, paramilitares, paramilitares, paramilitares, paramilitares, paramilitares, paramilitares, personas, personas, vecino]
1	Montes_de_maria.txt	[barrio, bolívar, bolívar, bolívar, cairo, caribe, carmen, carmen, carmen, carmen bolívar, carmen bolívar, cartagena, cartagena, casa, casa, casa, casa, casa, casa, casa, casa cultura, casa cultura, casa cultura, colombia, colombia, concejo municipal, escuela, ferretería, mercado viejo, montes maría, montes maría, montes maría, montes maría, montes maría, montes maría,	[asentaron, bomba, bomba, bomba, bomba, complot, conflicto, conflicto, conflicto, conflicto, desplazados, desplazados, desplazamiento, guerra, lucha, masacre, masacre,	[accidente, asesinados, carencia, memoria, memoria, memoria, memoria, miedo, miedo, miedo, muertos, muertos, noche, noche negra,	[agosto, febrero, julio, lunes, mayo, miércoles, octubre, septiembre]	[alcalde, amigas, amigos, amigos, autodefensas unidas colombia, beatriz, beatriz, beatriz, bloque norte, chichi, chichi, civiles, colectivo, colectivo, colectivo, colectivo, colectivo, colectivo, colectivo, colectivo, colombiano, comunicadores, concejales, dueño, farc, frente 37, fuerzas armadas colombia, fulano, guerrilla, guerrilla, guerrilla, guerrilla, guerrilla, hermano, hermano, hermano, jorge, maestros, mamá, muertos, muertos, museo itinerante memoria, museo

A su vez, para efectos de establecer las relaciones, se redujeron las frecuencias de las dimensiones a un solo elemento.

Figura 37

Dataset de palabras por dimensión

	Ubicaciones	Hecho Victimizante	Afectación	Temporalidad	Entidades
0	[bellavista, carretera, carrillo, casa, casa cural, casas, centro salud, chocó, colegio, colombia, departamento, iglesia, mares, medellín, mosquera, municipio, necoclí, noroeste, orilla, panamericana, región, restaurante, río, universidad, urgencias, vigía fuerte, zona]	[bomba, cilindro, cilindro bomba, combate, conflicto, daba, destrozo, explosión, masacre, matar, pipeta, refugiados]	[cáncer, esquirlas, herida, heridos, muerto, necesario, salud, secuelas]	[marzo, mayo, octubre]	[abuela, aliciente, antún ramos, auria, autoridades, auxiliar, bloque 58, camilo, civiles, comandante, enfermera, farc, guerrilla, joven, mamá, menores, misioneras agustinas, monjas, muerto, niña, paciente, padre, papá, paramilitares, personas, vecino]
1	[barrio, bolívar, cairo, caribe, carmen, carmen bolívar, cartagena, casa, casa cultura, colombia, concejo municipal, escuela, ferretería, mercado viejo, montes maría, municipio, parque, paz, país, porvenir, pueblo, región, salado, sucre, vereda, zona]	[asentaron, bomba, complot, conflicto, desplazados, desplazamiento, guerra, lucha, masacre, matar, secuestrar, sevicia, toque queda]	[accidente, asesinados, carencia, memoria, miedo, muertos, noche, noche negra, quemaduras]	[agosto, febrero, julio, lunes, mayo, miércoles, octubre, septiembre]	[alcalde, amigas, amigos, autodefensas unidas colombia, beatriz, bloque norte, chichi, civiles, colectivo, colombiano, comunicadores, concejales, dueño, farc, frente 37, fuerzas armadas colombia, fulano, guerrilla, hermano, jorge, maestros, mamá, muertos, museo itinerante memoria, paracos, paramilitares, personas, policía, sijin, soraya bayuelo, tía]

Este proceso permitió identificar 493 elementos, de los cuales 451 se encuentran asociados a las dimensiones, así: 138 en Ubicaciones, 67 en Hechos victimizantes, 62, 24 y 160 en las dimensiones Afectaciones, Temporalidad y Entidades, respectivamente. Los restantes 42 elementos corresponden a 7 nombres de los archivos y 35 dimensiones, pues las 5 dimensiones se encuentran en todos los testimonios.

Figura 38

Distribución de frecuencia de tokens por dimensión

	Archivo	Ubicaciones	Ubic. Únicas	Hechos	H. Únicas	Afectaciones	Afect. Únicas	Temporal	Temp. Únicas	Entidades	Ent. Únicas
0	Bojaya.txt	61	27	16	12	15	8	11	3	54	26
1	Montes_de_maria.txt	57	26	23	13	15	9	8	8	61	31
2	Segovia.txt	47	23	24	12	24	13	8	5	52	31
3	Trujillo.txt	37	21	19	14	24	17	3	3	73	32
4	Cauca.txt	31	17	4	4	6	4	0	0	21	5
5	Sintetico.txt	9	3	1	1	2	2	0	0	2	2
6	Bahia_portete.txt	70	21	16	11	16	9	9	5	76	33

El proceso de identificación de similitud entre textos se efectuó con un *corpus* más delimitado, pues de las 52.266 palabras cargadas inicialmente, el proceso se realizó con 7.731, es decir, considerando solo el 14,79% de las palabras.

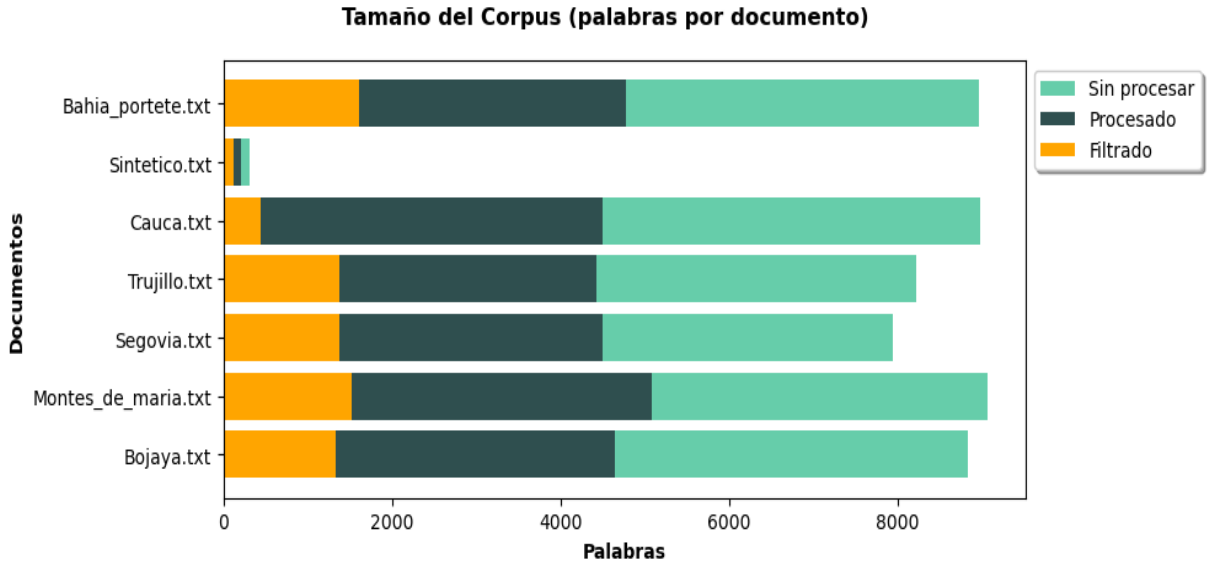
Tabla 4

Comparativo de palabras entre corpus inicial preprocesado y filtrado

	Archivo	Testimonio base	Testimonio limpio	Testimonio filtrado
0	Bojaya.txt	8822	4635	1327
1	Montes_de_maria.txt	9059	5081	1511
2	Segovia.txt	7932	4498	1373
3	Trujillo.txt	8211	4422	1377
4	Cauca.txt	8974	4498	432
5	Sintetico.txt	309	208	110
6	Bahia_portete.txt	8959	4774	1601

Figura 39

Tamaño del corpus, palabras por documento



La aplicación de la técnica TF-IDF (considerando los parámetros 1, 3, como se mencionó en el capítulo anterior), tuvo como resultado una matriz de 7 filas y 1.632 columnas, por lo que se puede argumentar que se tuvo en cuenta el contexto de las palabras.

Figura 40

TF-IDF de corpus normalizado

	11	11 noche	11 noche domingo	12	12 abril	12 abril febrero	37	37 fuerzas	37 fuerzas armadas	58	...	zona brazo	zona brazo calcinado	zona zona	zona zona asentaron	zona zona bomba	zona zona brazo	zona zona zona	éxodo	éxodo accidente	éxo acciden asesinad
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	...	0.00	0.00	0.02	0.00	0.03	0.00	0.00	0.00	0.00	0.
1	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.03	0.00	...	0.00	0.00	0.04	0.03	0.00	0.00	0.03	0.00	0.00	0.
2	0.03	0.03	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
3	0.00	0.00	0.00	0.03	0.03	0.03	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.02	0.02	0.01	0.00	0.00	0.02	0.00	0.02	0.02	0.

7 rows × 1632 columns

Por su parte, al calcular las distancias se logró identificar una alta cantidad de similitud entre los testimonios *Segovia.txt* y *Sintetico.txt*, ya que la función distancia coseno reporta 0,430825, esto debido a que en el archivo creado artificialmente se registraron elementos que hicieran coincidir con el testimonio en referencia. Por su parte, de manera automática, dos de los

documentos que también presentaron similitud fueron *Bojaya.txt* y *Montes_de_Maria.txt* con una distancia de 0,208267. Las demás relaciones entre los testimonios se representan en la siguiente tabla, asociando en las filas y columnas los números de los archivos asignados en los *dataframes* presentados más arriba.

Tabla 5
Distancia entre documentos del corpus

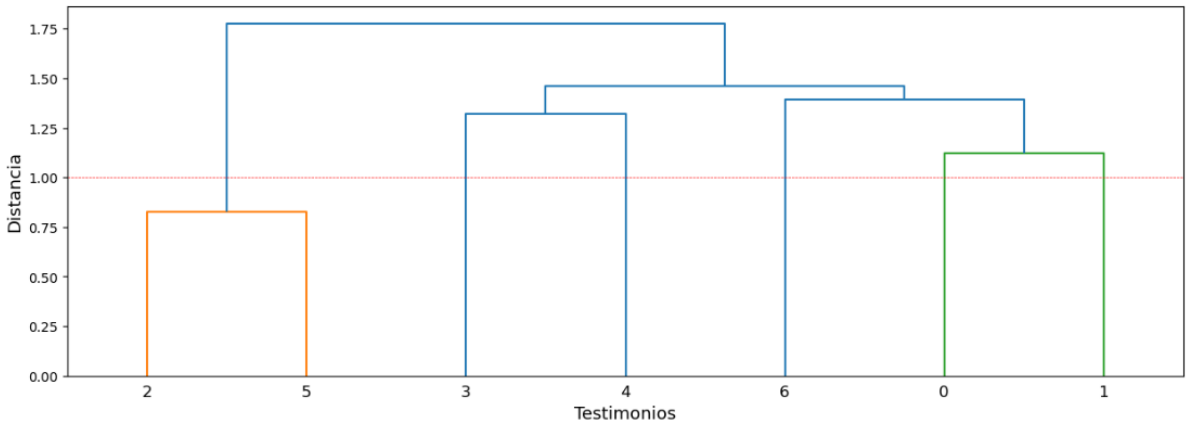
	0	**1**	**2**	**3**	**4**	**5**	**6**
0	1,000000	0,208267	0,116921	0,050531	0,066732	0,022669	0,032190
1	0,208267	1,000000	0,125055	0,060430	0,050402	0,005789	0,103144
2	0,116921	0,125055	1,000000	0,073544	0,056693	0,430825	0,062897
3	0,050531	0,060430	0,073544	1,000000	0,065406	0,007621	0,008395
4	0,066732	0,050402	0,056693	0,065406	1,000000	0,004165	0,015559
5	0,022669	0,005789	0,430825	0,007621	0,004165	1,000000	0,007152
6	0,032190	0,103144	0,062897	0,008395	0,015559	0,007152	1,000000

En lo referente a la representación gráfica de la similitud, se pueden identificar 6 grupos, el primero por los documentos etiquetados con los códigos 2 y 5, correspondientes a los textos *Segovia.txt* y *Sintetico.txt*, como se apuntó en el párrafo anterior; a su vez, el siguiente grupo, conformado por los testimonios *Bojaya.txt* y *Montes_de_Maria.txt*, etiquetados con los códigos 0 y 1, pues guardan similitud al tener varios tokens coincidentes; el tercer grupo está formado por los documentos que representan los testimonios *Trujillo* y *Cauca*, etiquetados con los códigos 3 y 4. Los siguientes 3 grupos están formados por la unión del documento restante, *Bahia_Portete.txt*, el cual está etiquetado con el código 6 y asociado con el grupo o clúster 2 que, a su vez, se integra al grupo 3 para generar el quinto grupo, el cual se vincula en un último grupo con el clúster número 1. Las siguientes figuras evidencian las representaciones gráficas encontradas.

Tabla 6
Agrupación de testimonios por similitud

	Enlace A	Enlace B	Distancia	Tamaño de clúster
0	2.0	5.0	0.825372	2.0
1	0.0	1.0	1.122244	2.0
2	3.0	4.0	1.321985	2.0
3	6.0	8.0	1.392766	3.0
4	9.0	10.0	1.460579	5.0
5	7.0	11.0	1.775192	7.0

Figura 41
Dendograma de clústeres jerárquico



Finalmente, el grafo (resultado de las relaciones identificadas previamente) presenta una red de puntos o nodos en color azul, los cuales se hacen menos intensos a medida que baja el nivel, es decir, los elementos de nivel 1 (documentos) tienen un color más intenso que los de nivel 3 (elementos de las dimensiones). Los enlaces o líneas que interconectan los nodos representan las relaciones identificadas, por lo que es posible notar relaciones entre elementos comunes de distintos documentos, por ejemplo, la generada por el término “Violencia” (**Figura 42**); la relación entre elementos y dimensiones, como por ejemplo, entre la palabra anterior y la dimensión Afectación

(**Figura 43**); y la relación entre las dimensiones y documentos, como la efectuada entre la dimensión Afectación y el documento Sintético.TXT (**Figura 44**).

Figura 42
Detalle de relaciones de nivel 3 entre la dimensión Violencia de dos documentos

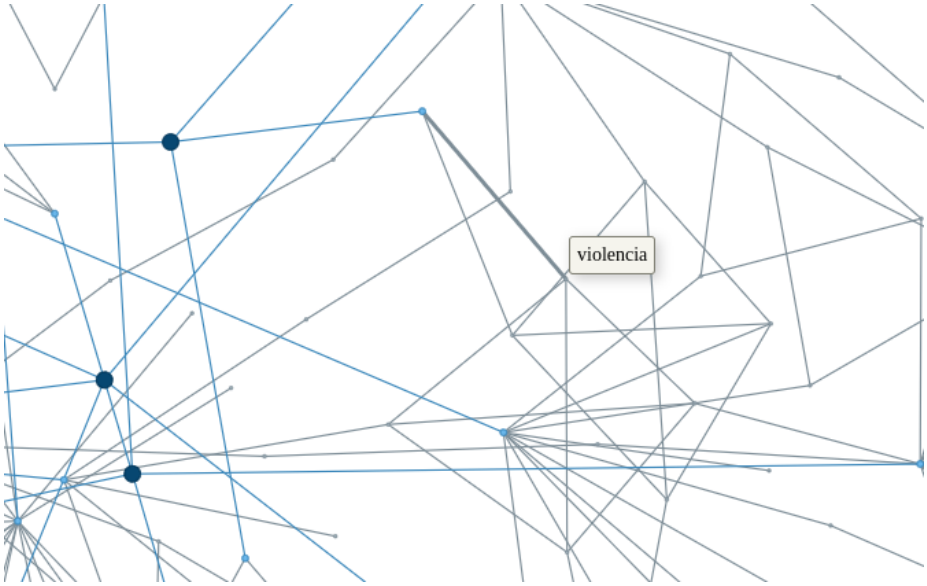


Figura 43
Detalle de la relación entre la dimensión Afectación y el elemento Violencia

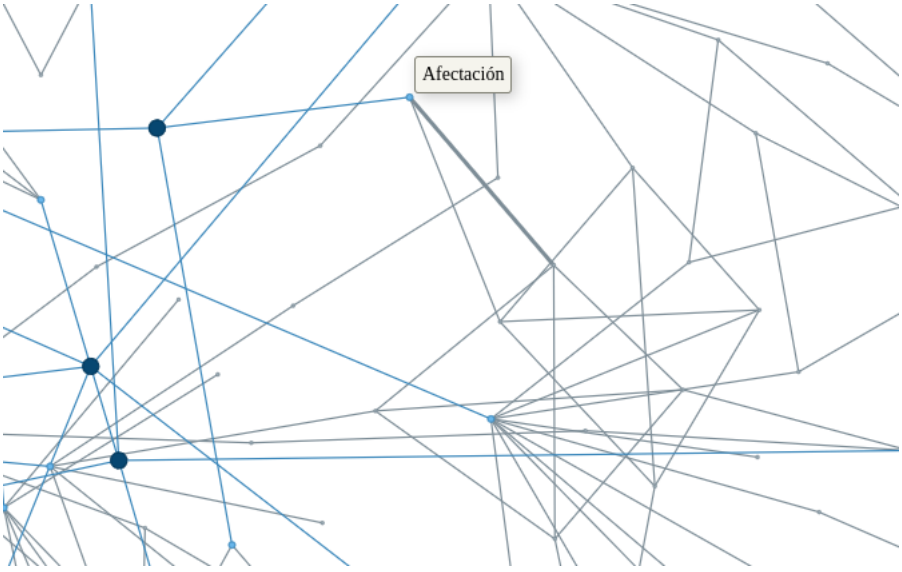
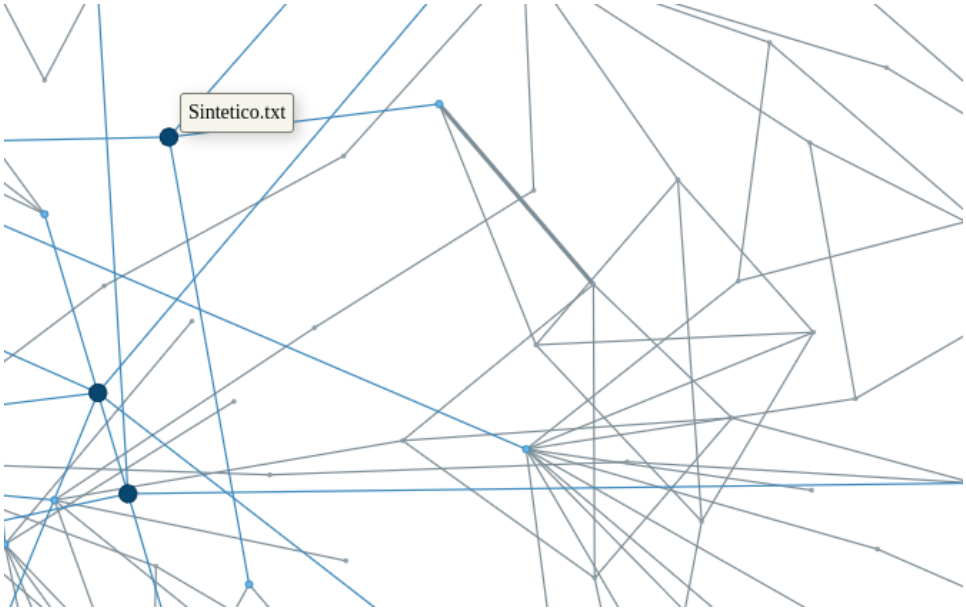


Figura 44

Detalle de la relación entre el documento Sintético.txt y la dimensión Afectación

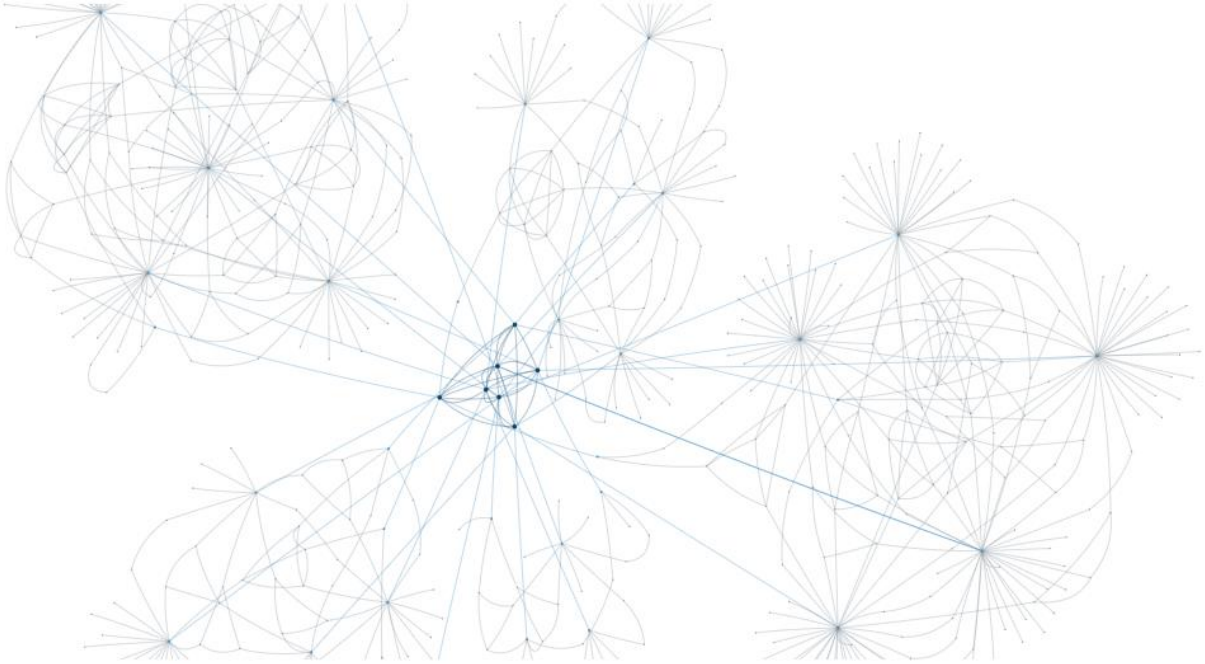


La figura se presenta en un documento HTML interactivo, el cual permite aplicar *zoom in* y *zoom out* sobre el lienzo, para permitir mayor nivel de detalle sobre cada elemento. Asimismo, presionando clic sobre un nodo es posible detallar el nombre y las relaciones con otros elementos.

Esta representación gráfica de los documentos, sus dimensiones y elementos, permite identificar que hay una relación entre todos los testimonios analizados, pues si bien es cierto que no todos narran los mismos hechos, existe coincidencia entre las dimensiones encontradas. De igual forma, el análisis visual permite destacar que las dimensiones “Entidades” y “Ubicaciones” son las que más elementos contienen, ratificando lo expresado previamente en la **figura 38**; por tanto, al ser las dimensiones con mayor representación de elementos por documento, se logra evidenciar que por lo menos un elemento en cada documento tiene coincidencia con su semejante en al menos otro documento. Esto se ejemplifica con la relación del elemento "Niña" del documento Bahia_Portete.txt, el cual se encuentra presente en la dimensión "Entidades" tanto de este documento como de *Bojayá.txt* y *Trujillo.txt*.

Figura 45

Representación gráfica de las relaciones de documentos, dimensiones y términos



7 Conclusiones

Es relevante destacar que los documentos del *corpus* están compuestos por 8.659 caracteres en promedio, alcanzando un total de 52.266; sin embargo, al realizar el preprocesamiento y la normalización, presentan una reducción del 46,2%; finalmente, resultan 7.731 caracteres, los cuales componen los elementos coincidentes en las dimensiones. Es decir, el algoritmo de relacionamiento y similitud de documentos se realizó considerando el 14,8% del *corpus* inicial, el cual puede aumentar en la medida que nuevas palabras no consideradas se sumen en las dimensiones.

El análisis de los tokens revela que los documentos del *corpus* contienen 9502 palabras, y durante el preprocesamiento se eliminó un 62,25% de los tokens, lo que equivale a 5915 palabras, resultando en un conjunto final de 3587 palabras para todos los documentos. Posteriormente, en la selección de los elementos de las dimensiones, el *corpus* experimentó una nueva reducción de 2591 palabras respecto a la fase anterior, o de 8506 en comparación con el número inicial, lo que representa una disminución del 72,23% y 89,52% respectivamente. Finalmente, las actividades posteriores se realizaron con 996 tokens.

Por su parte, a pesar de contar solo con 309 caracteres iniciales, y luego de realizar los procesos de preprocesamiento y normalización, el documento sintético pasa a la ejecución del algoritmo con 110 caracteres, 14 palabras que identifican dimensiones y 8 vectores o palabras únicas. De manera intencional, no anexa palabras en la dimensión de temporalidad.

En lo relacionado con las dimensiones, se evidencia una notable diferencia entre las ubicaciones y las demás, ya que esta representa el 75,4% del total, comparado con el 6,1% promedio de las otras 4 dimensiones. Esta señalada desigualdad radica en que, como se indicó en secciones anteriores, la fuente de sus datos depende de estructuras consolidadas por instituciones gubernamentales; sin embargo, tanto esta dimensión como las otras, pueden ir en aumento a medida que nuevos elementos producto de análisis sean incluidas por investigadores que las integren.

Las líneas de código que presentan resúmenes de los datos tanto de *stopwords* como de las dimensiones, se presentan como punto de apoyo al investigador, pues ante la duda se constituyen en herramientas para confirmar si una palabra está incluida en alguno de estos elementos, y en el caso de no estar y requerirse, anexarla, bien sea en el código del preprocesador (*stopwords*) como en cualquiera de las 5 dimensiones.

Con relación a la identificación de los elementos comunes entre testimonios, se logró determinar que los documentos cuentan con valores en todas las dimensiones, a excepción de los archivos Sintetico.txt (por ser de manera intencional) y Cauca.txt, el cual no recuperó coincidencias para la dimensión de temporalidad. A su vez, se resalta que el documento Cauca.txt (en la dimensión hechos victimizantes) y los documentos Montes_de_maria.txt y Trujillo.txt (en la dimensión temporal) tienen la misma cantidad de elementos y vectores, lo que quiere decir que la frecuencia para cada elemento coincidente con tales dimensiones es igual a 1 (**Figura 46**).

Figura 46
Relación de identificación de elementos por frecuencia y únicos por dimensión y documento

	Archivo	Ubicaciones	Ubic. Únicas	Hechos	H. Únicas	Afectaciones	Afect. Únicas	Temporal	Temp. Únicas	Entidades	Ent. Únicas
0	Bojaya.txt	61	27	16	12	15	8	11	3	54	26
1	Montes_de_maria.txt	57	26	23	13	15	9	8	8	61	31
2	Segovia.txt	47	23	24	12	24	13	8	5	52	31
3	Trujillo.txt	37	21	19	14	24	17	3	3	73	32
4	Cauca.txt	31	17	4	4	6	4	0	0	21	5
5	Sintetico.txt	9	3	1	1	2	2	0	0	2	2
6	Bahia_portete.txt	70	21	16	11	16	9	9	5	76	33

Igualmente, como se apuntó previamente, el dendograma permite evidenciar la agrupación de los documentos en 6 enlaces; sin embargo, destacan 3 clústeres formados por los pares de documentos Sintetico.txt y Segovia.txt; Bojayá.txt y Montes_de_maria.txt; y Trujillo.txt y Cauca.txt; principalmente, por compartir elementos en las dimensiones basadas en la ubicación geográfica, tipos de relatos y afectaciones narradas, así como la frecuencia de los elementos coincidentes.

Así mismo, es importante referenciar que el desarrollo del algoritmo utilizó diversas técnicas de PLN, tanto en la fase de preprocesamiento y limpieza de textos, con la implementación de las funciones de corrección de ortografía, lematización, normalización de caracteres en minúsculas, y eliminación de espacios en blanco, caracteres especiales y palabras vacías o *stopwords*; así como en la fase de identificación de elementos comunes con las funciones de extracción de características e identificación de similitud, tales como TF-IDF y *linkage*.

Finalmente, y gracias a la codificación de los elementos comunes en los testimonios, las relaciones se proyectan en el grafo en *Pyvis*, el cual, de manera dinámica, permite hacer un seguimiento o trazabilidad a las relaciones, rastreando las conexiones y los nodos que tiene un documento con otro.

8 Recomendaciones

Como se evidencia durante la investigación, a pesar de que el número de testimonios es limitado por los motivos previamente expuestos, los objetivos de establecer las relaciones entre los documentos y la identificación de la similitud entre los mismos, se alcanzó con éxito. Sin embargo, es recomendable considerar un mayor número de textos o testimonios para validar tanto el desempeño del algoritmo como la identificación de nuevos patrones o elementos ocultos entre líneas.

De igual forma, es importante considerar la ampliación de los componentes de cada dimensión, pues de esta manera es posible identificar un mayor número de elementos entre los documentos y, por tanto, la detección de coincidencias entre ellos, lo que aumenta la precisión del modelo automático frente al mismo procedimiento que realiza un investigador. La construcción de diccionarios más completos o bancos de términos más robustos para las dimensiones puede ser, incluso, de gran utilidad para las ciencias sociales, al proponer la consolidación de ontologías especializadas en la materia.

A su vez, aunque el algoritmo logra establecer relaciones y similitudes entre los documentos, se considera interesante aplicar diferentes técnicas, tanto tradicionales como de vanguardia, a fin de establecer validaciones del método y considerar mayor precisión al tener en cuenta de manera más precisa la semántica de los textos.

Por otra parte, aun contando con una prudente representación gráfica de los elementos del *corpus* y algoritmo (tales como grafos de redes sociales, dendogramas y gráficos de barras), es conveniente considerar nuevas técnicas que puedan aportar a una mayor visualización y estética de los resultados.

Finalmente, puesto que una de las motivaciones de este trabajo es el de aportar y ser útil a actividades motivadas en el proceso de paz entre la guerrilla de las FARC y el Gobierno de Colombia, se considera relevante la difusión, tanto del resultado de la investigación como de su algoritmo pues, como se demostró, a través de la identificación de los elementos, la consolidación

de tablas y la representación a través de grafos, es posible recuperar información oculta que solo se logra vislumbrar si se establecen las relaciones entre los documentos.

Referencias

Al Omran, F. N. A., & Treude, C. (2017). Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments. *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, 187-197. https://doi.org/10.1109/MSR.2017.42

Arhab, N., Oussalah, M., & Jahan, S. (2022). Social media analysis of car parking behavior using similarity based clustering. *Journal of Big Data*, *9*(1), 74. https://doi.org/10.1186/s40537-022-00627-x

Barrón-Cedeño, A., Rosso, P., Agirre, E., & Labaka, G. (2010). *Plagiarism detection across distant language pairs*. *2*, 37-45. Scopus.

Bird, S., Klein, Ewan, & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners*. https://doi.org/10.48550/ARXIV.2005.14165

Colombia. Centro Nacional de Memoria Histórica. (2019). *La vida cuenta: Pódcasts de memoria*. https://bit.ly/4ci7BDi

Colombia. Centro Nacional de Memoria Histórica. (2020). *Pódcasts de memoria*. https://bit.ly/3PkvGiZ

Colombia. Centro Nacional de Memoria Histórica. Grupo de Memoria Histórica. (2013). *¡Basta ya! Colombia: Memorias de guerra y dignidad. Informe general Grupo de Memoria Histórica* (p. 432). Centro Nacional de Memoria Histórica.

Colombia. Centro Nacional de Memoria Histórica. Observatorio de Memoria y Conflicto. (2020). *Categorías*. https://bit.ly/3PpIp43

Colombia. Comisión de la Verdad. (2022). *Hay futuro si hay verdad: Informe final de la Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición (11 tomos en 24 volúmenes)*. Comisión de la Verdad.

Colombia. Comisión de la Verdad. (2023). *Analítica de datos (información y recursos): Transcripciones de entrevistas (anonimizadas)*. https://bit.ly/4ccv20V

Colombia. Comisión de la Verdad. (2024a). *Analítica de datos (información y recursos): Documentos y recursos de analítica de datos creados y utilizados por la Comisión para la investigación del Informe Final, disponibles para ampliar el estudio sobre el conflicto armado colombiano*. https://bit.ly/3IB7TaR

Colombia. Comisión de la Verdad. (2024b). *¿Qué es la comisión de la verdad?* http://comisiondelaverdad.co

Colombia. Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición. (2022). *Sistematización y analítica de datos*. Comisión de la Verdad.

Colombia. Congreso de la República. (2005). *Ley 975 de 2005 (julio 25): Por la cual se dictan disposiciones para la reincorporación de miembros de grupos armados organizados al margen de la ley, que contribuyan de manera efectiva a la consecución de la paz nacional y se dictan otras disposiciones para acuerdos humanitarios*. Diario Oficial.

Colombia. Congreso de la República. (2017). *Acto Legislativo 01 (4 de abril de 2017): Por medio del cual se crea un tí-tulo de disposiciones transitorias de la Constitución para la terminación del conflicto armado y la construcción de una paz estable y duradera y se dictan otras disposiciones*. Diario Oficial.

Colombia. Ministerio de Tecnologías de la Información y las Comunicaciones. (2024). *Departamentos y municipios de Colombia*. https://bit.ly/49RCrB4

Colombia. Presidencia de la República. (2017). *Decreto 588 de 2017 (abril 5): Por el cual se organiza la Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición*. Diario Oficial.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*. https://doi.org/10.48550/ARXIV.1810.04805

Duque Bedoya, E. T. (2009). *Metodología para la extracción de metadatos semánticos de textos en español utilizando procesamiento de lenguaje natural: Subaplicación para la identificación de contextos espaciales y temporales en textos que describan interacciones*

entre actores. [Tesis de maestría, Universidad EAFIT]. Repositorio Institucional Universidad Eafit. http://hdl.handle.net/10784/1261

Flores, M. Á. (2019). *Implementando un corrector ortográfico en Python, utilizando la distancia de Levenshtein*. https://bit.ly/3WdC3cf

Godoy Viera, Á. F. (2017). Técnicas de aprendizaje de máquina utilizadas para la minería de texto. *Investigación Bibliotecológica*, *31*(71), 103-126. https://doi.org/10.22201/iibi.0187358xp.2017.71.57812

Gomaa, W. H., & Fahmy, A. F. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications, 68*(13), 13-18. https://doi.org/10.5120/11638-7118

Goutte, C., & Gaussier, E. (2005). *A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation*. Advances in Information Retrieval, Santiago de Compostela, España.

Hogenboom, F., Frasincar, F., Kaymak, U., De Jong, F., & Caron, E. (2016). A Survey of event extraction methods from text for decision support systems. *Decision Support Systems*, *85*, 12-22. https://doi.org/10.1016/j.dss.2016.02.006

Jurafsky, D., & Martin, J. H. (2014). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall.

Ko, B., Choi, D., Choi, C., Choi, J., & Kim, P. (2012). Document classification through building specified N-gram. *6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012*, 171-176. https://doi.org/10.1109/IMIS.2012.142

Le, Q., & Mikolov, T. (2013). Distributed Representations of Sentences and Documents. *Proceedings of the 31 St International Conference on Machine Learning*.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of Massive Datasets*. Stanford University.

Mahmoud, A., & Zrigui, M. (2020). Semantic Similarity Analysis for Corpus Development and Paraphrase Detection in Arabic. *The International Arab Journal of Information Technology, 18*(1), 1-7. https://doi.org/10.34028/iajit/18/1/1

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Mariani, J., Francopoulo, G., & Paroubek, P. (2018). Reuse and plagiarism in Speech and Natural Language Processing publications. *International Journal on Digital Libraries*, *19*(2-3), 113-126. https://doi.org/10.1007/s00799-017-0211-0

Maxwell, J. C. (2022). *Las 21 leyes irrefutables del liderazgo*. Grupo Nelson.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Neural and Information Processing System (NIPS): 26*.

Montes y Gómez, M. (2005). Minería de texto empleando la semejanza entre estructuras semánticas. *Computación y Sistemas*, *9*(1), 63-81.

Nawar, A., Toma, N. T., Al Mamun, S., Kaiser, M. S., Mahmud, M., & Rahman, M. A. (2021). Cross-Content Recommendation between Movie and Book using Machine Learning. *2021 IEEE 15th International Conference on Application of Information and Communication Technologies (AICT)*, 1-6. https://doi.org/10.1109/AICT52784.2021.9620432

Parodi, G. (2008). Lingüística de corpus: Una introducción al ámbito. *RLA. Revista de Lingüística Teórica y Aplicada*, *46*(1), 93-119.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543. https://doi.org/10.3115/v1/D14-1162

Sarkar, D. (2019). *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing*. Apress. https://doi.org/10.1007/978-1-4842-4354-1

Speechnotes. (2024). *Speechnotes: Speech to Text—Voice Typing & Transcription* [Software]. Speechnotes. https://speechnotes.co/

Srinivasarao, U., & Sharaff, A. (2023). Spam email classification and sentiment analysis based on semantic similarity methods. *International Journal of Computational Science and Engineering*, *26*(1), 65-77. https://doi.org/10.1504/IJCSE.2023.129147

Sugiyama, K., & Kan, M. Y. (2015). A comprehensive evaluation of scholarly paper recommendation using potential citation papers. *International Journal on Digital Libraries*, *16*(2), 91-109. https://doi.org/10.1007/s00799-014-0122-2

Sumathi, S., & Sivanandam, S. N. (2006). *Introduction to Data Mining and its Applications* (Vol. 29). Springer. https://doi.org/10.1007/978-3-540-34351-6

Vallejo-Huanga, D., Jaime, J., & Andrade, C. (2023). Similarity Visualizer Using Natural Language Processing in Academic Documents of the DSpace in Ecuador. En I. Sserwanga, A. Goulding, H. Moulaison-Sandy, J. T. Du, A. L. Soares, V. Hessami, & R. D. Frank (Eds.), *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity* (pp. 343-359). Springer. https://doi.org/10.1007/978-3-031-28032-0_28

Anexos

- Archivos de audio en formato MP3, descargados del micrositio Pódcasts del Centro Nacional de Memoria Histórica (CNMH).

- Archivos de transcripciones de audios descargados del Centro Nacional de Memoria Histórica (CNMH).

- "Transcripciones de entrevistas (anonimizadas)", resultado del proceso de escucha de la Comisión para el Esclarecimiento de la Verdad, la Convivencia y la No Repetición (CEV), consolidación de una carpeta de 2.500 archivos de texto que contienen testimonios de víctimas del conflicto armado en Colombia.

- Listado de dimensiones en formato csv.

- Notebook Jupyter con código en código escrito en Python de relacionamiento de testimonios aplicado en la investigación.