UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

ANÁLISIS DE DESPLAZAMIENTO SEMÁNTICO PREVIO Y POSTERIOR AL COVID-19 EN *WORD EMBEDDINGS* DIACRÓNICOS DEL ESPAÑOL

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del Programa de Estudios de Posgrado en Computación e Informática para optar al grado y título de Maestría Profesional en Computación e Informática

ESTEBAN RODRÍGUEZ BETANCOURT

Ciudad Universitaria Rodrigo Facio, Costa Rica

2022

Dedicatoria

Le dedico este trabajo a mi mamá, Maritza Betancourt Alvarado, por su apoyo incondicional y estar siempre presente.

Agradecimientos

Le agradezco al Dr. Edgar Casasola Murillo por su guía durante este proceso del TFIA. También a Dra. Gabriela Marín Raventós y Dra. Kryscia Ramírez Benavides por sus aportes durante la revisión de este trabajo.

Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Computación e Informática para optar al grado y título de Maestría Profesional en Computación e Informática.

Firmado digitalmente por JORGE ANTONIO LEONI DE LEON (FIRMA)
Fecha: 2022.06.29 12:08:52 -06'00'

Dr. Jorge Antonio Leoni de León

Representante de la Decana del Sistema de Estudios de Posgrado

EDGAR ENRIQUE CASASOLA MURILLO (FIRMA)
PERSONA FISICA, CPF-01-0729-0430.
Fecha declarada: 28/06/2022 07:20:03 PM
Esta representación visual no es fuente de confianza. Valide siempre la firma.

Dr. Edgar Casasola Murillo

Director

GABRIELA MARIN RAVENTOS (FIRMA)

Firmado digitalmente por GABRIELA MARIN RAVENTOS (FIRMA)
Fecha: 2022.06.28 18:45:31 -06'00'

Dra. Gabriela Marín Raventós

Asesora

KRYSCIA RAMIREZ BENAVIDES (FIRMA)

Firmado digitalmente por KRYSCIA RAMIREZ BENAVIDES (FIRMA)
Fecha: 2022.06.29 08:31:37 -06'00'

Dra. Kryscia Ramírez Benavides

Asesora

GUSTAVO LOPEZ HERRERA (FIRMA)

Firmado digitalmente por GUSTAVO LOPEZ HERRERA (FIRMA)
Fecha: 2022.06.29 13:55:58 -06'00'

Dr. Gustavo López Herrera

Representante de la Directora del Programa de Posgrado en Computación e Informática

ESTEBAN RODOLFO RODRIGUEZ BETANCOURT (FIRMA)

Firmado digitalmente por ESTEBAN RODOLFO RODRIGUEZ BETANCOURT (FIRMA)
Fecha: 2022.06.28 19:00:11 -06'00'

Esteban Rodríguez Betancourt

Sustentante

Índice general

Resumen

El significado de las palabras puede cambiar a lo largo del tiempo (Hamilton, Leskovec, y Jurafsky, 2016). Este fenómeno se conoce como desplazamiento semántico. Existen diferentes formas de medir este desplazamiento semántico, siendo una de ellas el análisis de los cambios en distancias en los *embeddings* de las palabras.

En este trabajo, se propone un mecanismo para la construcción de *word embeddings* diacrónicos, es decir, de diferentes momentos del tiempo. Luego se construye un *word embedding* diacrónico del español previo y posterior a la aparición de la pandemia por COVID-19. Estos *embeddings* fueron construidos a partir de un corpus 237 millones de sitios web. Finalmente, se analiza el desplazamiento semántico de los términos asociados a tres casos de estudio particulares: COVID-19, vacunación y mascarillas.

A pesar de que pasaron pocos años entre la recolección de los corpus con los que se entrenaron los *word embeddings*, se encontraron cambios significativos en las vecindades de los clústeres de palabras analizadas. Además, se encontraron cambios sutiles en la distancia relativa a emociones.

Abstract

Word meanings can change through time (Hamilton et al., 2016). This behavior is known as semantic shift. There are several ways to measure this shift, one is analyzing the distance changes between word embeddings.

This work proposes a mechanism for building diachronic word embeddings: word embeddings that consider time. Then, a diachronic word embedding model for Spanish is built, with data before and after the surge of COVID-19 pandemic. These word embeddings were built from a corpus with 237 million documents from web sites. Finally, the semantic shift of terms related with COVID-19, vaccination and masks is analyzed.

Although there were just a few years between both training corpus, significant changes in the neighborhood of analyzed clusters were found. Also, subtle changes in distance relative to emotions were found.

Índice de figuras

Índice de tablas

Índice de Abreviaturas

ASIC: *Application-specific integrated circuit*. Es un circuito integrado diseñado para un uso específico.

AWS: *Amazon Web Services*

BERT: *Bidirectional Encoder Representations from Transformers* (Devlin, Chang, Lee, y Toutanova, 2019). Es un modelo de *word embeddings* donde las palabras tienen una representación densa diferente según su contexto.

CLD2: *Compact Language Detector 2*. Es un clasificador bayesiano para detectar el idioma de un texto.

CPU: *Central Processing Unit*.

CSS: *Cascading Style Sheets*.

DBSCAN: *Density-based spatial clustering of applications with noise*. Es un algoritmo de agrupamiento.

EC2: *Amazon Elastic Compute Cloud*. Plataforma de cómputo en la nube de AWS.

EPS: ϵ (épsilon). Distancia máxima entre los elementos de un clúster, al usar DBSCAN.

GPU: *Graphics Processing Unit*. Tipo de coprocesador especializado en operaciones gráficas, como multiplicaciones de matrices. Debido a esto es común su uso en aprendizaje de máquina.

HTTP: *Hypertext Transfer Protocol*.

JSON: *JavaScript Object Notation*. Es un formato para intercambio de datos.

S3: *Amazon Simple Storage Service*. Es un servicio de AWS para almacenar archivos.

TB: Terabyte.

TFIA: Trabajo Final de Investigación Aplicada.

TPU: *Tensor Processing Unit*. Es un tipo de ASIC especializado en operaciones de tensores, usadas en algoritmos de aprendizaje de máquina.

T-SNE: *T-distributed Stochastic Neighbor Embedding*. Es un método estadístico para visualizar datos de alta dimensionalidad en dos o tres dimensiones.

URL: *Uniform Resource Locator*. La dirección de un recurso en internet, por ejemplo, un sitio web.

USD: *United States Dollar*.

WARC: *Web ARChive*. Es el formato usado por el *Internet Archive* y *CommonCrawl* para almacenar sus descargas.

UNIVERSIDAD DE COSTA RICA

SEP Sistema de Estudios de Posgrado

Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Esteban Rodríguez Betancourt, con cédula de identidad 1-1451-0124, en mi condición de autor del TFG titulado Análisis de desplazamiento semántico previo y posterior al COVID-19 en word embeddings diacrónicos del español

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. **SI** [X] **NO** *[ ]

***En caso de la negativa favor indicar el tiempo de restricción: _____ año (s).**

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

INFORMACIÓN DEL ESTUDIANTE:

Nombre Completo: Esteban Rodríguez Betancourt.

Número de Carné: B15512 Número de cédula: 1-1451-0124.

Correo Electrónico: _____.

Fecha: 10 de junio del 2022. Número de teléfono: _____.

Nombre del Director (a) de Tesis o Tutor (a): Edgar Casasola Murillo.

ESTEBAN RODOLFO RODRIGUEZ BETANCOURT (FIRMA)

Firmado digitalmente por ESTEBAN RODOLFO RODRIGUEZ BETANCOURT (FIRMA)
Fecha: 2022.06.28 19:07:41 -06'00'

FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no sólo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

Capítulo 1

Introducción

Los *word embeddings* se han convertido en una de las principales herramientas para representar texto en el área de procesamiento de lenguaje natural, al asociar una palabra con una representación vectorial de su «significado» (Mikolov, Chen, Corrado, y Dean, 2013). Estas representaciones pueden ser usadas posteriormente en tareas de recuperación de información, clasificación, generación de texto y otras (Mikolov, Yih, y Zweig, 2013).

Los *word embeddings* se construyen a partir de grupos de documentos conocidos como corpus (Sierra, 2017). Debido a que el significado de las palabras es dinámico a través del tiempo (Hamilton et al., 2016), la cercanía entre ellas puede variar según la época del corpus con el que se entrena el modelo de *embeddings*. En la actualidad existen investigaciones donde se han estudiado estos cambios de significado, incluso cambios semánticos a través de décadas. Por ejemplo, Hamilton et al. (2016) estudiaron los cambios en las palabras en inglés entre el año 1800 y 2009. Estos cambios de significado en el tiempo, conocidos como *desplazamiento semántico*, pueden ser visualizados al analizar la distancia vectorial existente entre los *word embeddings* de las palabras.

Según Hamilton et al. (2016) los cambios en las palabras siguen las siguientes leyes:

1. Ley de conformidad: La tasa de cambios semánticos es proporcional a una potencia negativa de la frecuencia de las palabras.

2. Ley de innovación: Las palabras más polisémicas tienen mayores tasas de cambio semántico.

Debido a estos cambios en significados se han propuesto modelos de *word embeddings* diacrónicos que consideran la información temporal. Yao, Sun, Ding, Rao, y Xiong (2017) propusieron un método de *word embeddings* condicionado también por la ubicación temporal. Gong, Bhat, y Viswanath (2020) propusieron un modelo de *embeddings* calculado en función del tiempo y el lugar (además de la relación entre palabras), con la intención de poder determinar tendencias culturales o situaciones específicas de un lugar.

Para este TFIA se propone analizar los cambios en las distancias entre palabras, al comparar dos modelos de *word embeddings* entrenados con corpus en español recolectados en diferentes años, antes y después de la aparición del COVID-19. Para lograrlo se entrenarán dos modelos de *word embeddings* con dos corpus tomados de internet abierto anteriores y posteriores a la aparición del COVID-19. Posteriormente, estos *embeddings* serán alineados utilizando una adaptación de la técnica de alineamiento bilingüe supervisado propuesta Joulin, Bojanowski, Mikolov, Jégou, y Grave (2018). Ya con los *embeddings* alineados es posible realizar medir las distancias de coseno entre los dos vectores de cada palabra o encontrar cambios de similitud (acercamientos o alejamientos) entre palabras.

A continuación se justifica por qué es necesario estudiar cómo los cambios en significado de las palabras se pueden ver reflejados en los *word embeddings*. Posteriormente, se definen los objetivos de la investigación y las limitaciones de esta. En el Capítulo 2 en la página 6 se mencionan investigaciones que han usado las técnicas que se pretenden usar en este TFIA, trabajos similares realizados anteriormente y en qué se diferencian con la investigación actual. En el Capítulo 3 en la página 8 se describen conceptos relacionados con la presente investigación, como *web crawler* o *word embeddings*. En el Capítulo 4 en la página 14 se describe la metodología del trabajo y cómo se realizará cada uno de los objetivos específicos. También se detallan los recursos computacionales que serán utilizados en la investigación. Los resultados se muestran en varias secciones del Capítulo 5 en la página 26. En la Apartado 5.1 en la página 26 se detalla cómo se elaboró el corpus diacrónico y cómo se construyeron los *word embeddings*. En la Apartado 4.5 en la página 17 se detalla cómo se midió el desplazamiento semántico y el desplazamiento emocional. En el Apartado 5.2 en la página 34 se analiza el desplazamiento semántico de varios casos de estudio respecto a las palabras más cercanas y en el Apartado 5.3 en la página 67 se analiza el desplazamiento de los casos de estudio respecto a palabras asociadas a emociones. El Capítulo 6 en la página 73 indica las conclusiones del trabajo y posibles áreas de mejora y trabajo futuro.

1.1. Justificación

Actualmente, los modelos de aprendizaje profundo y *word embeddings* son utilizados en muchas aplicaciones, como búsqueda (Li, Qin, Wang, Chen, y Metzler, 2020) o detección de discursos de odio (Navarro-Murillo, Calvo-Vargas, y Casasola-Murillo, 2019). Es razonable pensar que estos modelos con el paso del tiempo pierdan efectividad y deban ser entrenados de nuevo con datos actualizados. Pero, ¿esto puede suceder en pocos años?

Los *word embeddings* muchas veces no son entrenados para cada aplicación específica, sino que se reutiliza algún modelo precalculado. Esto facilita enormemente su uso, pero surge la interrogante de si el modelo precalculado escogido realmente funciona para la tarea a realizar. Ya es un hecho conocido que los *word embeddings* contienen y amplifican la parcialización presente en los datos de entrenamiento, como prejuicios y estereotipos (Papakyriakopoulos, Hegelich, Serrano, y Marco, 2020). Por otro lado, un modelo debería ser capaz de generalizar y adaptarse bien a nuevas situaciones. ¿Pero cómo se comportan los *word embeddings* ante nueva información totalmente desconocida?

Conocemos como *distribution shift* (desplazamiento distribucional) a los cambios de distribución estadística entre los datos sobre los que fue entrenado un modelo de aprendizaje de máquina y los datos sobre los que se ejecuta (Balaji, 2021). Estas diferencias pueden reducir la efectividad de un modelo de aprendizaje de máquina e incluso pueden ser fatales si suceden en sistemas críticos, como vehículos autónomos o diagnóstico médico (Balaji, 2021).

Situaciones que causan grandes cambios sociales, como la pandemia del COVID-19, previsiblemente pueden cambiar la forma en que nos expresamos. Se han realizado estudios recientes sobre cómo esto afecta el significado de las palabras sobre *Tweets* en inglés (Guo, Xypolopoulos, y Vazirgiannis, 2021) o bien se han desarrollado *embeddings* específicos sobre documentos médicos relacionados con el COVID-19 (Miranda-Escalada et al., 2021). Sin embargo, **no se encontraron estudios sobre el cambio en la similitud de los *embeddings* en español** con documentos tomados de internet abierto, ni con diferencias de pocos años.

Existen diferentes razones que justifican la realización de esta investigación. La más directa es una observación del movimiento entre conceptos relacionados con temas que fueron impactados por la pandemia. A nivel computacional, el aporte consiste en utilizar los *word embeddings* diacrónicos como herramienta para automatizar el análisis de información.

Otra razón de interés es que el estudio pretende identificar cambios en un período corto de tiempo producto de un evento global como lo es la pandemia. El corpus diacrónico es el primero para español creado con este fin, lo que hace que esta propuesta tenga un factor importante de novedad.

Finalmente, otro potencial beneficio a nivel computacional consiste en identificar los períodos de obsolescencia en un *embedding* producto del cambio en los corpus que lo generan en dominios específicos. Dicho de otro modo, estudiar los cambios en similitud de las palabras respecto a sus *embeddings* puede ayudar a determinar cuando es necesario volver a entrenar un modelo, dado que el re-entrenamiento implica costo. Modelos recientes de *word embeddings* como BERT (Devlin et al., 2019) son costosos de entrenar y en algunos casos esto solamente es posible para centros de investigación de la industria con acceso a hardware especializado y amplios recursos económicos (Izsak, Berchansky, y Levy, 2021).

Este trabajo es pionero en Costa Rica, ya que no existe otro estudio a nivel nacional que haya propuesto y construido *word embeddings* diacrónicos en español para el análisis de los desplazamientos semánticos que se han dado producto del COVID-19.

1.2. Objetivos

A continuación se presenta el objetivo general y los objetivos específicos planteados para este trabajo.

Objetivo general

Analizar el desplazamiento semántico en *word embeddings* diacrónicos en español.

Objetivos específicos

1. Proponer un mecanismo para la construcción de un *word embedding* diacrónicos.

2. Construir un *word embedding* diacrónico de español previo y posterior a la pandemia por COVID-19.

3. Analizar el desplazamiento semántico de términos asociados a tres casos de estudio particulares, utilizando el *word embedding* diacrónico alineado.

Alcances y limitaciones

Los objetivos anteriores tendrán algunas restricciones, relacionadas con limitaciones en acceso a recursos computacionales o bien a datos precalculados.

Los *word embeddings* serían entrenados utilizando un corpus de documentos en español extraídos de internet abierto. Se utilizará el conjunto de datos recolectado por el proyecto CommonCrawl [1] para realizar un entrenamiento con documentos existentes en un punto del 2018 y otro punto del 2021. Inicialmente, se pretende usar el algoritmo *word2vec* (Mikolov, Chen, et al., 2013) para la generación de los *word embeddings*, ya que este corre eficientemente en CPU tradicionales, que son el tipo de equipo con el que se cuenta. No obstante, según el costo económico del procesamiento, se seleccionaría otra implementación de este algoritmo, como por ejemplo *BlazingText* (Gupta y Khare, 2017) que se ejecuta en GPU o *pWord2Vec* (Ji, Satish, Li, y Dubey, 2016).

Idealmente, se hubieran utilizado otros algoritmos de *word embeddings* más recientes, como por ejemplo BERT (Devlin et al., 2019). Sin embargo, esto es económicamente inviable, debido al elevado costo de entrenarlo. Según la cantidad de parámetros, entrenar BERT puede tener un costo entre $2500 USD y $1600000 USD y además requiere hardware especializado como TPU (Sharir, Peleg, y Shoham, 2020). Por otro lado, correr otro algoritmo de *embeddings* puede ser significativamente más barato. Por ejemplo, de forma anecdótica, anteriormente el investigador ha entrenado modelos de *word embeddings* utilizando *BlazingText* (Gupta y Khare, 2017) para un corpus de medio millón de documentos y el costo total no ha excedido los $15 USD.

En el siguiente capítulo se describirán algunos trabajos similares al presente TFIA y otras investigaciones sobre cambio semántico.

[1] `https://commoncrawl.org/about`

Capítulo 2

Estado del arte

Los *word embeddings* son representaciones de las palabras usando vectores de números reales. Estas representan el significado de una palabra, tal que palabras similares tienen un *embedding* o vector similar (Jurafsky y Martin, 2000). Debido a que las palabras cambian de significado, es esperable que estos cambios sean detectados en sus *word embeddings*. Esto ya ha sido estudiado por otros investigadores, como por ejemplo Hamilton et al. (2016), quienes propusieron las leyes de Conformidad y de Innovación. Este estudio fue realizado sobre textos de libros en inglés, escritos entre 1800 y 2009.

Los cambios semánticos han sido estudiados con diversas técnicas a lo largo del tiempo. Kutuzov, Øvrelid, Szymanski, y Velldal (2018) menciona que antes de los modelos de *word embeddings* estos cambios eran principalmente estudiados utilizando la frecuencia de las palabras. Según Kutuzov et al. (2018), Kulkarni, Al-Rfou, Perozzi, y Skiena (2014) publicaron el primer estudio que demostraba que usar las representaciones de *word embeddings* podía dar mejores resultados que los métodos usando frecuencias. Kulkarni et al. (2014) usaron *word embeddings* para determinar los cambios semánticos de algunas palabras a través del tiempo, como por ejemplo *gay* o *apple*. Kutuzov et al. (2018) menciona que anteriormente otros autores habían utilizado técnicas similares para estudiar los cambios semánticos, pero en una escala menor o sin demostrar que este tipo de estudios daba mejores resultados que los basados en frecuencias.

Las técnicas tradicionales de *word embeddings*, como *word2vec*, se pueden convertir en *embeddings* diacrónicas. Para lograr tener *embeddings* diacrónicos Montariol (2021) menciona que se han usado dos grupos de técnicas: alinear vectores no dia-

crónicos o bien crear un modelo de vectores que dentro de su construcción considere la dimensión de tiempo.

Más recientemente, Montariol (2021) propone un método para medir el cambio semántico en *word embeddings* contextuales, como BERT (Devlin et al., 2019). Este método sería capaz de detectar cambios semánticos en palabras con diferentes significados según el contexto. Sin embargo, el costo de entrenar BERT lo hace prohibitivo para la presente investigación.

Específicamente sobre COVID-19, Miranda-Escalada et al. (2021) presentaron un modelo de *word embeddings* con fastText usando *tweets* en español que contenían palabras asociadas al COVID-19. Sin embargo, este no es un *embedding* diacrónico y no se investigaron los cambios semánticos en español.

Asif, Zhiyong, Iram, y Nisar (2020) estudiaron los cambios causados por el COVID-19 en el idioma inglés. Este estudio se enfocó en neologismos y los cambios fueron observados manualmente.

Butler y Simon-Vandenbergen (2021) realizaron un estudio del cambio semántico causado por el COVID-19 en palabras en inglés. En este caso los cambios semánticos fueron medidos usando la frecuencia de las palabras y no *word embeddings*, que es la técnica que se desea utilizar en este trabajo.

Guo et al. (2021) investigaron los cambios semánticos relacionados con el COVID-19 de las palabras en inglés, a partir de *tweets*. De las investigaciones encontradas esta sería la más similar a la actual. No obstante, se diferencia en que la propuesta actual se pretende realizar en español y utilizando un corpus recolectado de internet abierto en lugar de *tweets*.

En el siguiente capítulo se describen algunos términos y modelos utilizados en esta investigación.

Capítulo 3

Marco Teórico

Este TFIA se va a concentrar en las diferencias entre dos modelos de *word embeddings*. Sin embargo, se requiere crear primero un corpus del 2021, que será usado para entrenar uno de los modelos de *word embeddings*. Para recolectar este corpus se requiere una araña.

3.1. Araña web o *web crawler*

Un *web crawler* o araña es un sistema que realiza el proceso de recolectar sitios web para indexarlos y soportar un motor de búsqueda (Manning, 2008). Típicamente, una araña comienza con una o más URL que constituyen el conjunto semilla. Estos documentos son descargados, parseados y se le extraen los hipervínculos, con lo cual el proceso se repite de forma recursiva (Manning, 2008). En este caso se utilizarán dichos documentos para entrenar un modelo de *word embeddings*.

3.2. CommonCrawl

Para este trabajo se requieren elaborar *word embeddings* a partir de documentos recolectados en dos momentos diferentes: antes del COVID-19 y posterior a la aparición de este. Para propósitos de este TFIA, empezado a mediados del 2021, claramente sería imposible ejecutar una araña web en el pasado. Sin embargo, existen proyectos como Web Archive[1] o CommonCrawl [2] que se han dedicado a recolectar

[1]Disponible en `http://web.archive.org/`
[2]Disponible en `https://commoncrawl.org/`

vastas cantidades de sitios web y además están agrupados cronológicamente.

CommonCrawl es una organización que «construye y mantiene un repositorio abierto de datos de arañas web que pueden ser accedidas y analizadas por cualquiera». Esta organización descarga regularmente (típicamente una vez al mes) grandes cantidades de documentos de internet y los resultados los hace disponibles al público en general. Debido al gran volumen de datos y a que se tienen datos históricos, esta es una fuente sumamente valiosa para realizar un estudio como el que se propone en este trabajo. CommonCrawl expone los datos descargados principalmente de tres formas: los datos crudos, el texto limpio (sin etiquetas HTML) y metadatos.

Los datos son guardados en formato WARC y comprimidos con GZip. El formato WARC permite añadir ciertos metadatos, como encabezados HTTP. Los metadatos también son publicados dentro de archivos WARC, como un JSON. Cada archivo WARC tiene una gran cantidad de documentos, la cual varía según cada archivo, pero suelen ser unos 40000 documentos. Cada corpus tiene entre 64000 y 72000 archivos WARC y cada archivo WARC mide aproximadamente 200 MB (comprimido).

Aparte del contenido de los sitios web, CommonCrawl publica metadatos computados a partir de los documentos. Algunos de estos metadatos son el idioma detectado y la codificación detectada. En el caso del idioma, se indica el porcentaje de texto en cada idioma, según lo calcula el algoritmo CLD2[3].

3.3. *Word embeddings*

Los *word embeddings* son una representación vectorial del «significado» de una palabra. Aunque previamente han existido representaciones vectoriales, como *one-hot encoding*, en este caso consideramos *word embeddings* a las representaciones densas, que reducen el significado a cada palabra a una cantidad fija de dimensiones (por ejemplo, 300 o 100 dimensiones), tal como se conceptualiza en el trabajo de Mikolov, Sutskever, Chen, Corrado, y Dean (2013).

Estas representaciones vectoriales presentan propiedades interesantes, que son útiles para hacer análisis de texto natural. Por ejemplo, Bengio, Ducharme, Vincent, y Janvin (2003) describen cómo palabras con significado similar suelen ser cercanas en el espacio vectorial. En el trabajo de Mikolov, Yih, y Zweig (2013) se muestran ejemplos de resolución de analogías utilizando los *word embeddings*. Este tipo de resolución de analogías se ilustra en la ecuación (3.1):

[3]https://github.com/CLD2Owners/cld2

$$V[king] - V[man] + V[woman] \sim V[queen] \tag{3.1}$$

En la ecuación anterior se muestra cómo los *word embeddings* pueden usarse para resolver analogías. En este caso, *man* (hombre) es a *woman* (mujer) como *king* (rey) es a *queen* (reina). En este caso, *queen* fue la palabra con el *word embedding* más cercano al vector resultante de las operaciones aritméticas sobre los *word embeddings*.

Existen muchos algoritmos para asociar una palabra a un *word embedding*. Uno de los más conocidos es *Word2Vec* (Mikolov, Chen, et al., 2013), el cual utiliza la capa oculta de una red neuronal para representar cada palabra. Dicha red es entrenada para identificar la palabra oculta dado el contexto (*CBOW*) o el contexto dada la palabra (*Skipgram*). También está GloVe (Pennington, Socher, y Manning, 2014), el cual aprovecha la información estadística de las palabras. Otro modelo es *fastText* (Joulin, Grave, Bojanowski, y Mikolov, 2017), que entre otras cosas usa información de subpalabras, lo que le permite generar *embeddings* para palabras desconocidas.

También hay mejoras en el rendimiento computacional. Con *pWord2Vec*, Ji et al. (2016) paralelizaron *word2vec* para arquitecturas con muchos núcleos y memoria RAM de rápido acceso entre núcleos. *BlazingText* (Gupta y Khare, 2017), se basó en las técnicas mostradas en *pWord2Vec* y lo adaptaron para ejecutar *word2vec* y *fastText* en GPU y de forma distribuida.

Los modelos anteriores asocian una palabra a un único vector. Sin embargo, según el contexto las palabras pueden tener significados totalmente diferentes. Existen modelos como *Universal Sentence Encoding* (Cer et al., 2018) y *Bidirectional Encoder Representations from Transformers* (*BERT*) (Devlin et al., 2019) que son capaces retornar *embeddings* diferentes dependiendo del contexto en el que se utilice la palabra.

3.4. *Word embeddings* diacrónicos

Las palabras pueden cambiar su significado a través del tiempo, reflejando cambios tanto en la lengua, como en la sociedad (Kutuzov et al., 2018). Una forma de analizar estos cambios es mediante *word embeddings* diacrónicos (Kutuzov et al., 2018), los cuales son *word embeddings* generados a partir de un corpus diacrónico. Un corpus diacrónico es un corpus que contiene documentos de varios momentos del tiempo, tal y como lo explica Sierra (2017).

Anteriormente, ya se han realizado estudios de desplazamiento semántico a través de muchas décadas usando *word embeddings*. Por ejemplo, Hamilton et al. (2016)

analizaron 200 años de textos en Inglés, Francés, Alemán y Mandarín usando *word2vec*, entre otras técnicas. Entre sus aportes está la técnica de alineamiento, la descripción de la técnica para medir el cambio, la visualización de los cambios de significado de las palabras y la propuesta de leyes estadísticas sobre el cambio semántico.

Hamilton et al. (2016) usan el algoritmo de Procrustes ortogonales para matrices propuesto por Schönemann (1966) para alinear los *word embeddings* diacrónicos Hamilton et al. (2016). Con esto lograron preservar los ángulos entre los vectores y hacerlos comparables, ya que la transformación es una rotación. Para determinar el desplazamiento de las palabras usaron la medida de correlación de Spearman sobre una serie generada con la distancia de coseno entre pares de la misma palabra a través del tiempo. Además, para visualizar los desplazamientos semánticos en el tiempo usaron una proyección de T-SNE con las palabras y sus vecinos más cercanos en cada paso de tiempo.

Dados *word embeddings* alineados, Hamilton et al. (2016) definen el desplazamiento semántico entre dos periodos como DistanciaCoseno $(w_t, w_{t+\Delta})$, donde w_t representa el vector correspondiente al *word embedding* de la palabra w en el momento t. En general, la similitud de coseno de dos vectores se define como se muestra en la Ecuación (3.2) y la distancia de coseno de dos vectores se define tal como se muestra en la Ecuación (3.3):

$$\text{SimilitudCoseno}\,(a, b) = \frac{a \cdot b}{|a| \cdot |b|} \tag{3.2}$$

$$\text{DistanciaCoseno}\,(a, b) = 1 - \text{SimilitudCoseno}\,(a, b) \tag{3.3}$$

3.5. Clasificación de emociones

Para clasificar las emociones se optó por utilizar la clasificación jerárquica propuesta por Shaver et al. (1987). Esta consiste en varias emociones primarias, que se dividen en emociones secundarias y estas se dividen en emociones terciarias, tal como se muestra en la Tabla 3.1 en la página 13.

Para realizar la clasificación automática de emociones predominantes en una oración, Alshahrani, Samothrakis, y Fasli (2017) utilizaron *Word Mover's Distance* para calcular la distancia entre listados de palabras y una oración. Esta es una técnica no supervisada que es capaz de predecir las emociones presentes en un texto.

Word Mover's Distance (Kusner, Sun, Kolkin, y Weinberger, 2015) es una técnica

para medir la distancia entre dos documentos u oraciones. Básicamente, se relaciona cada palabra en un documento con alguna palabra en el otro, de forma que la suma de la distancia de coseno entre cada par de palabras se minimice.

En el siguiente capítulo se detalla la metodología utilizada para lograr los objetivos de la investigación.

Primaria	Secundaria	Terciaria
Love	*Affection*	*Adoration, fondness, liking, attraction, caring, tenderness, compassion, sentimentality*
	Lust	*Desire, passion, infatuation*
	Longing	*Longing*
Joy	*Cheerfulness*	*Amusement, bliss, gaiety, glee, jolliness, joviality, joy, delight, enjoyment, gladness, happiness, jubilation, elation, satisfaction, ecstasy, euphoria*
	Zest	*Enthusiasm, zeal, excitement, thrill, exhilaration*
	Contentment	*Pleasure*
	Optimism	*Eagerness, hope*
	Enthrallment	*Enthrallment, rapture*
	Relief	*Relief*
Surprise	*Surprise*	*Amazement, astonishment*
Anger	*Irritability*	*Aggravation, agitation, annoyance, grouchy, grumpy, crosspatch*
	Exasperation	*Frustration*
	Rage	*Anger, outrage, fury, wrath, hostility, ferocity, bitterness, hatred, scorn, spite, vengefulness, dislike, resentment*
	Disgust	*Revulsion, contempt, loathing*
	Envy	*Jealousy*
	Torment	*Torment*
Sadness	*Suffering*	*Agony, anguish, hurt*
	Sadness	*Depression, despair, gloom, glumness, unhappiness, grief, sorrow, woe, misery, melancholy*
	Disappointment	*Dismay, displeasure*
	Shame	*Guilt, regret, remorse*
	Neglect	*Alienation, defeatism, dejection, embarassment, homesickness, humiliation, insecurity, insult, isolation, loneliness, rejection*
	Sympathy	*Pity, mono no aware, sympathy*
Fear	*Horror*	*Alarm, shock, fear, fright, horror, terror, panic, histeria, mortification*
	Nervousness	*Anxiety, suspense, uneasiness, apprehension, worry, distress, dread*

Tabla 3.1: Jerarquía de emociones propuesta por Shaver et al. (1987)

Capítulo 4

Metodología

En este capítulo se describe la metodología utilizada para lograr los objetivos planteados en este TFIA. Esta investigación se relaciona con el curso de Procesamiento de Lenguaje Natural y Recuperación de Información. La metodología está resumida en la Figura 4.1 en la página 22.

A nivel general, se usó un proceso metodológico basado en las ciencias del diseño. En la primera iteración se hizo una aproximación inicial al problema y se envió un artículo a la conferencia CLEI 2021, adjuntado en el Apéndice D en la página 97, el cual no fue aceptado. Los comentarios recibidos fueron incorporados en la siguiente iteración del trabajo, el cual derivó en la presente iteración.

La metodología para lograr los objetivos específicos se describe en las siguientes secciones.

4.1. Mecanismo para la construcción de un *word embedding* diacrónico

Como parte del Objetivo 1, en esta sección se propone un mecanismo para realizar la construcción de un *word embedding* diacrónico. Este mecanismo debe solventar los siguientes aspectos:

1. Obtener un corpus diacrónico: se necesita un corpus con documentos anteriores y posteriores a la aparición del COVID-19.

2. Construir el modelo de *word embeddings*

3. Alinear los *word embeddings* generados, para obtener un *word embeddings* diacrónico.

4. Ser realizable con las restricciones de recursos (equipo y financiamiento) existente.

Los aspectos anteriores deben ser integrados en el mecanismo propuesto, el cual se muestra a alto nivel en la Figura 4.2 en la página 23.

El resultado de este objetivo específico es una metodología que describe paso a paso cómo generar un *word embedding diacrónico* y su respectiva implementación.

Una vez completada esta etapa se procedió a crear un modelo de *word embeddings diacrónicos* y aplicar el modelo para detectar cambios semánticos entre los dos años (2018 y 2021).

4.2. Construcción de un *word embedding* diacrónico de español previo y posterior a la pandemia por COVID-19

Para la construcción de un *word embedding* diacrónico de español previo y posterior a la pandemia por COVID-19 fue necesario realizar varias tareas, las cuales se resumen en la Figura 4.3 en la página 24.

Para el objetivo específico 2 se construyó un modelo de *word embeddings* diacrónicos. Para obtener este modelo fue necesario construir un corpus diacrónico a partir de una fuente de datos, en este caso CommonCrawl. Posteriormente se generó un modelo de *word embeddings word2vec* para cada corpus. Una vez calculados los modelos de *word embeddings* se usó la técnica de Procrustes ortogonales para matrices propuesta por Schönemann (1966) para alinear los *word embeddings* diacrónicos, tal como realizó Hamilton et al. (2016). Esta es una técnica que sirve para alinear dos matrices usando rotaciones, lo cual preserva los ángulos entre los vectores. Además, es completamente no supervisada, por lo que no se requieren datos etiquetados adicionales.

La propuesta de alinear los *embeddings* reduce significativamente la cantidad de cálculos necesarios para detectar desplazamientos semánticos, pues reduce la complejidad computacional a $O(n)$, en lugar de $O(n^2)$ que sería necesaria para calcular distancias relativas entre cada par de palabras.

El producto de esta etapa fue un *word embedding* diacrónico alineado que puede ser usado para medir desplazamientos semánticos.

4.3. Análisis del desplazamiento semántico de casos de estudio particulares, utilizando el *word embedding* diacrónico alineado

Para analizar el desplazamiento semántico de casos de estudio particulares, utilizando el *word embedding* diacrónico alineado se identificaron las palabras con mayor desplazamiento semántico y estas fueron agrupadas de forma no supervisada. Posteriormente, se seleccionaron clústeres de interés, para los cuales se analizaron vecinos en el 2018, vecinos en el 2021, mayores acercamientos y mayores alejamientos. También se hizo un análisis de cercanía y desplazamiento relativo a palabras asociadas a emociones. Las tareas realizadas, resumidas en la Figura 4.4 en la página 24, fueron las siguientes tareas:

1. Identificación de las palabras con mayor desplazamiento semántico
2. Agrupamiento de las palabras de forma no supervisada a partir de los *word embeddings* de las palabras.
3. Selección manual de los clústeres de interés. Se descartaron clústeres compuestos por palabras sin sentido, errores ortográficos, código fuente, pocas o demasiadas palabras, similitud con otros clústeres, etc.
4. Determinar cuáles palabras fueron las más cercanas en el 2018 y en el 2021 a los clústeres estudiados.
5. Determinar cuáles fueron las palabras que más se alejaron o acercaron a los clústeres estudiados.
6. Para algunas palabras seleccionadas:
    a) Determinar cuáles palabras fueron las más cercanas en el 2018 y en el 2021 a la palabra en cuestión.
    b) Determinar cuáles fueron las palabras que más se alejaron o acercaron a la palabra en cuestión.

7. Medir el desplazamiento semántico de los clústeres respecto a emociones.

Por tratarse de un caso de estudio, al final se tendrá un análisis cualitativo del desplazamiento de las palabras y algunas estadísticas descriptivas relacionadas con las mismas.

4.4. Recursos

La realización de este TFIA requirió el uso de diversos recursos computacionales, los cuales se detallan en profundidad en sus respectivas secciones. El corpus de CommonCrawl está disponible públicamente y alojado en *Amazon Web Services S3*. Para aprovechar esta cercanía, el corpus fue descargado usando servidores de *EC2* en la nube de *Amazon Web Services*, en la misma región. Posteriormente, los *word embeddings* se calcularon usando la implementación de *BlazingText* disponible en *Amazon Sage Maker*. Finalmente, el análisis de los *word embeddings* resultantes fue realizado en la computadora de escritorio del autor, usando Go y Python.

4.5. Metodología para el Análisis del Desplazamiento Semántico

Para analizar el desplazamiento semántico de las palabras se hizo una selección de las palabras cuyos vectores en los dos periodos estudiados tuvieron una similitud de coseno inferior a 0,7 y aparecieron en el corpus al menos 1000 veces. Con este grupo de palabras se procedió a hacer un agrupamiento usando los *word embeddings*. Los clústeres resultantes fueron examinados manualmente y se seleccionaron los que correspondían a temas conocidos. Se excluyeron clústeres correspondientes a errores ortográficos, código fuente, ruido (como tiendas en línea o nombres de ciudades) y temas desconocidos o con mucha ambigüedad (cantantes de música árabe o actores coreanos, por ejemplo).

4.5.1. Agrupamiento

Para agrupar las palabras se emplearon varias pasadas del algoritmo DBSCAN (Ester, Kriegel, Sander, y Xu, 1996). Se comenzó con un EPS $= 0{,}65$. Todos los clústeres con más de 30 elementos fueron descartados y se repitió el algoritmo sobre sus

elementos, pero con un EPS de EPS $*0{,}9$. El proceso se detuvo si EPS $< 0{,}1$, quedaban menos de 5 palabras o bien el proceso ya se había repetido 25 veces.

Se decidió repetir el algoritmo de agrupamiento porque en el espacio de los *word embeddings* las palabras no tienen la misma densidad. Por este motivo, algunos clústeres interesantes desaparecían al usar un EPS bajo, pero, por otro lado, un EPS alto producía clústeres con muchos elementos.

4.5.2. Medición del Desplazamiento Semántico

Con la medición del desplazamiento semántico podemos averiguar qué tanto cambió el significado de una palabra de un momento a otro. Dados *word embeddings* alineados, Hamilton et al. (2016) definen el desplazamiento semántico como $\text{DistanciaCoseno}(w_t, w_{t+\Delta})$, donde w_t representa el vector correspondiente al *word embedding* de la palabra w en el momento t. En general, la similitud de coseno de dos vectores se define como se muestra en la Ecuación (4.1) y la distancia de coseno de dos vectores se define tal como se muestra en la Ecuación (4.2):

$$\text{SimilitudCoseno}(a, b) = \frac{a \cdot b}{|a| \cdot |b|} \tag{4.1}$$

$$\text{DistanciaCoseno}(a, b) = 1 - \text{SimilitudCoseno}(a, b) \tag{4.2}$$

La diferencia entre la similitud de coseno y la distancia de coseno es el codominio. En la similitud de coseno el rango es $[-1, 1]$, donde 1 representa que los vectores son iguales (tienen ángulo 0). Para la distancia de coseno el rango es $[0, 2]$, donde 0 representa los vectores iguales.

Con la distancia de coseno o la similitud de coseno es posible determinar cuáles fueron las palabras que cambiaron más. Sin embargo, dada la gran cantidad de palabras, no es práctico estudiar los desplazamientos individuales de cada palabra. Para reducir el problema se optó por analizar los desplazamientos con respecto a clústeres o agrupamientos de palabras. Para esto se detectaron las palabras con más desplazamiento semántico, se agruparon y se midió el desplazamiento semántico promedio dentro de cada clúster.

Para hacer comparaciones de qué tanto se acercó o alejó una palabra se definió una nueva métrica: «desplazamiento relativo». Dadas dos palabras a y b, un tiempo t y una diferencia de tiempo Δ, se define como desplazamiento relativo la Ecuación (4.3) en la página siguiente. Esta medida permite determinar si dos palabras se acercaron (valores son mayores a cero) o bien se alejaron (valores menores que cero).

$$\text{DesplazamientoRelativo}(a, b, t, \Delta) = 1 - \frac{\text{SimilitudCoseno}(a_{t+\Delta}, b_{t+\Delta}) + \epsilon}{\text{SimilitudCoseno}(a_t, b_t) + \epsilon} \quad (4.3)$$

En la Ecuación (4.3) se sumó un pequeño ϵ, $\epsilon = 10^{-5}$, para corregir posibles divisiones entre cero.

4.5.3. Medición de Similitud con Emociones

En algunos casos las palabras tienen mayor similitud o asociación con ciertas emociones que otras. Por ejemplo, si pensamos en la palabra «fiesta» es posible que la relacionemos con «alegría» o «sorpresa», pero no necesariamente con «temor» o «ira». Para medir la similitud de las palabras con una emoción es posible usar una técnica no supervisada, como *word2vec* y los ángulos entre los *word embeddings* resultantes. En esta sección, se presenta un mecanismo para medir estas relaciones. En la siguiente sección se usará esta técnica para determinar si hubo cambios en la similitud con ciertas emociones, por ejemplo, determinar si «vacuna» se acercó a «miedo» o «alivio».

Para medir la similitud emocional es necesario determinar qué emociones vamos a considerar. En este trabajo se decidió usar el agrupamiento jerárquico de emociones propuesto por Shaver et al. (1987). Cabe destacar que para crear esta clasificación los autores usaron listas de palabras en inglés y voluntarios evaluaron si la palabra era o no una emoción. Con esos datos, algoritmos de agrupamiento, conversión de las palabras a sustantivos y otras técnicas crearon una clasificación jerárquica. Al no encontrar una clasificación de emociones similar en Español, y siguiendo un criterio de conveniencia, para este trabajo se optó por traducir las palabras en las emociones terciarias. El utilizar texto traducido se considera una práctica común cuando no se dispone de recursos en un idioma diferente al inglés (Brooke, Tofiloski, y Taboada, 2009). Esto es una limitación de este trabajo y podría justificar la elaboración de un trabajo similar al de Shaver et al. (1987) específico para español.

En la Figura 4.5 en la página 25 se puede ver las emociones secundarias y terciarias según la clasificación de Shaver et al. (1987) para la emoción *love* (amor). Además, en el cuarto nivel del árbol se muestra la palabra base que se usó como traducción. Aparte de la palabra base se usaron modificaciones de la misma, como infinitivos, participios en ambos géneros (como «cuidado» y «cuidada»), gerundio, verbo reflexivo, sufijos «-ción» (como «adoración» o «atracción»), «-able» (como «adorable» o «compasible»). La lista completa de emociones y las traducciones usadas está

en el Apéndice B en la página 79.

Para medir la similitud de una palabra con alguna emoción en particular se midió la similitud de coseno entre la palabra estudiada y las traducciones correspondientes a la emoción que se está midiendo. Por ejemplo, si se desea cuantificar la similitud con «*love*» (amor) se mide la similitud de coseno entre la palabra y todas las palabras de las subemociones de *love*. De esta forma, es posible también medir el peso de subemociones específicas. Es decir, si E representa el conjunto de *word embeddings* de las palabras asociadas a una emoción y p es el *word embedding* de la palabra por estudiar, entonces la similitud con una emoción se define en la Ecuación (4.5) como:

$$\text{DistanciaEmocion}\,(p, E) = \min_{\forall e \in E} \text{DistanciaCoseno}\,(p, e) \tag{4.4}$$

$$\text{SimilitudEmocion}\,(p, E) = 1 - \text{DistanciaEmocion}\,(p, E) \tag{4.5}$$

La fórmula de «distancia a una emoción», en la Ecuación (4.4), puede ser considerada como un caso especial de usar *Word Mover's Distance* para medir la distancia a una emoción, tal como fue propuesto por Alshahrani et al. (2017). En el caso de Alshahrani et al. (2017) los autores usaron *Word Mover's Distance* y listados de emociones para determinar la emoción dominante en una oración. Sin embargo, en este caso, como solamente tenemos una palabra en la «oración» el algoritmo de *Word Mover's Distance* se reduce a simplemente buscar la distancia mínima entre esa única palabra y las emociones.

A los valores de similitud por emoción de todas las emociones se les puede aplicar softmax, obteniendo una distribución de probabilidad por emoción. En este trabajo, a este último resultado se llama medición de «emoción absoluta», para diferenciarlo de los desplazamientos semánticos relativos a emociones que se describirán en la siguiente sección.

Una limitación de esta medición es que no se realizaron experimentos para determinar la calidad de las asociaciones entre las palabras y las emociones, al no ser parte del alcance del presente trabajo. Sin embargo, la técnica no supervisada de usar *Word Mover's Distance* entre las palabras de una oración y términos que representan una emoción ya ha sido usada para determinar exitosamente las emociones de frases, tal como se menciona en Alshahrani et al. (2017), Alshahrani (2020) y Ren y Liu (2018).

4.5.4. Medición del Desplazamiento Semántico Relativo a Emociones

Para medir el desplazamiento semántico relativo a emociones simplemente se restan las emociones absolutas. De esta forma, se obtienen medidas que al sumarlas para todas las emociones suman cero. Así, es posible determinar qué emoción ganó o perdió peso respecto a una palabra.

4.5.5. Análisis de desplazamiento semántico

En el análisis del desplazamiento semántico de los clústeres estudiados se detalla una descripción del clúster (hecha a mano), la palabra más cercana al centroide del clúster, el promedio de similitud de coseno de las palabras en el clúster (entre 2018 y 2021), las palabras en el clúster, los vecinos más cercanos en 2018, vecinos más cercanos en 2021, las palabras que más se acercaron (usando la medida de «desplazamiento relativo»), las palabras que más se alejaron y una breve historia o explicación detrás del desplazamiento de algunas de las palabras del clúster. Además, se incluye una visualización realizada con una proyección t-SNE que muestra los vecinos más cercanos en 2018 y 2021, las palabras que más se acercaron y las que más se alejaron. En esta proyección se muestran flechas que indican la dirección del desplazamiento de cada palabra, las palabras del clúster se muestran en rojo, el centroide en azul y las palabras fuera del clúster en gris.

4.5.6. Selección de casos de estudio

Para evitar sesgos en la selección de los casos se decidió hacer la selección por criterios matemáticos: los grupos de palabras fueron agrupados automáticamente y ordenados según el desplazamiento semántico (de mayor desplazamiento a menor). Posteriormente, los clústeres fueron revisados manualmente para descartar grupos compuestos por errores de codificación, faltas ortográficas, código fuente (CSS o JavaScript), etc. También se excluyeron clústeres muy pequeños, ambiguos, con exceso de ruido o donde a conocimiento del autor no se pudiera extraer una relación.

Figura 4.1: Metodología

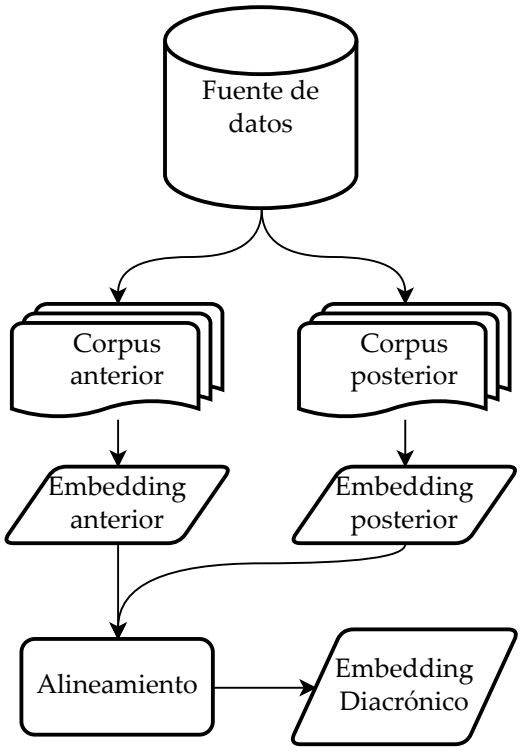

Figura 4.2: Metodología del objetivo específico 1

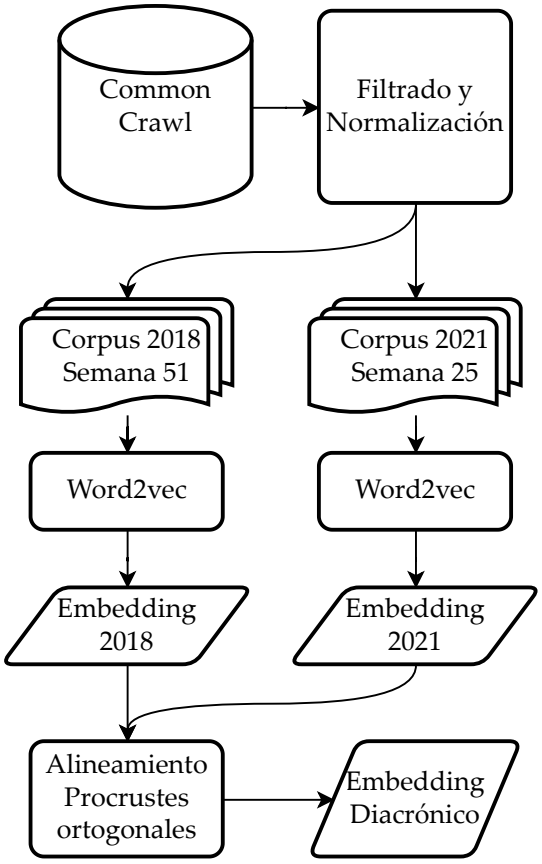

Figura 4.3: Metodología del objetivo específico 2

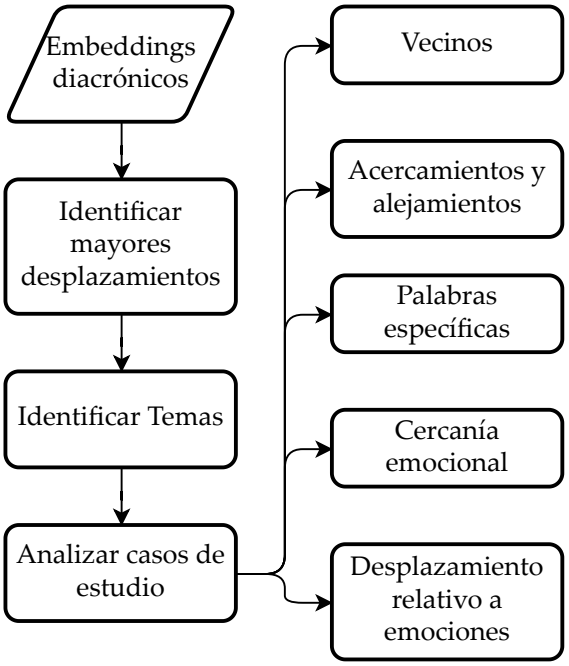

Figura 4.4: Metodología del objetivo específico 3

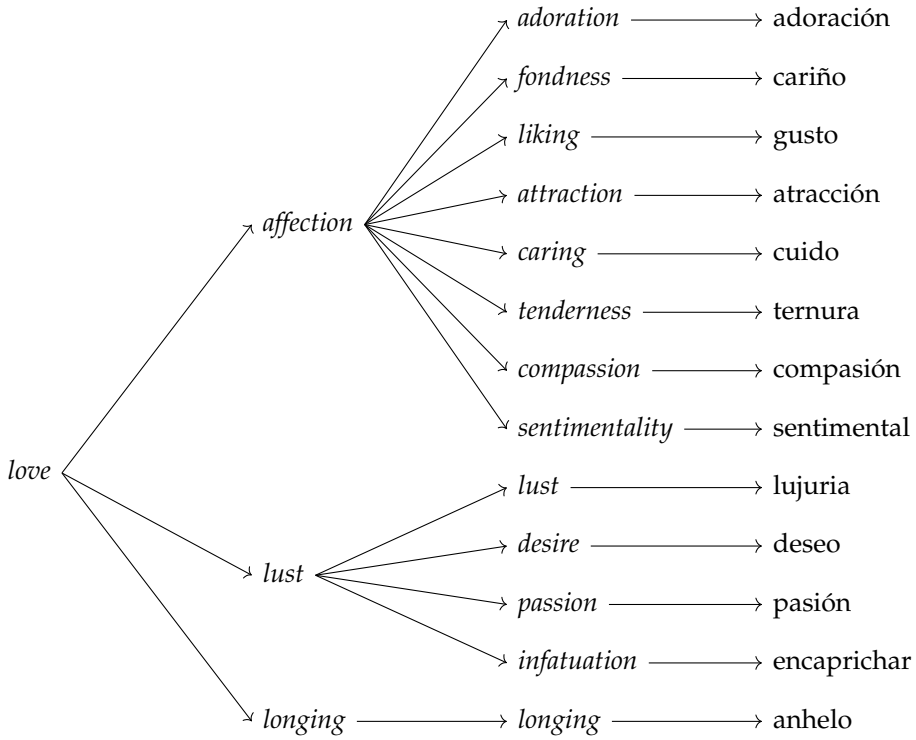

Figura 4.5: Emociones secundarias y terciarias bajo la emoción primaria *love* (amor) y la traducción usada

Capítulo 5

Resultados

En este capítulo se mostrarán y analizarán los resultados obtenidos, correspondientes a cada objetivo específico.

5.1. Mecanismo para la construcción de un *word embedding* diacrónico

En esta sección se detalla cómo realizar la construcción de un *word embedding* diacrónico. Para lograrlo hubo que resolver los siguientes aspectos:

1. Obtener un corpus de documentos recolectados en el 2018 y 2021 utilizando una araña web. Se utilizará el corpus disponible del proyecto CommonCrawl para esta etapa, ya que ellos cuentan con corpus históricos de años anteriores.

2. Construir *word embeddings* a partir del corpus recolectado en el 2018 y en el 2021. Para esto se usó el algoritmo *word2vec* con *skip-gramas*, específicamente la implementación *BlazingText*.

3. Alinear los *word embeddings* generados con el algoritmo de alineamiento no supervisado de Procrustes, usado por Hamilton et al. (2016).

4. Optimizar el mecanismo y su implementación para que sean realizables dentro de las restricciones existentes. Para lograrlo se comparó el costo de calcular *word2vec* con diferentes implementaciones y se optimizó la implementación para reducir el uso de recursos como memoria RAM o almacenamiento no volátil.

A continuación, se detalla cómo construir un modelo de *word embedding* diacrónico. Para esto es necesario:

- Seleccionar una fuente de documentos de las fechas requeridas (antes y después de la aparición del COVID-19)
- Descargar los datos
- Filtrar por idioma
- Convertir a UTF-8
- Normalizar el texto
- Eliminar duplicados
- Comprimir el corpus resultante
- Construir los *word embeddings*
- Alinear los *word embeddings*

5.1.1. Selección de fuente de los documentos

La generación de un corpus diacrónico es el primer paso necesario para construir *word embeddings* diacrónicos. Para construir este corpus es necesario descargar conjuntos de documentos de dos momentos diferentes en el tiempo: por ejemplo, en este trabajo se usarán conjuntos del 2018 y 2021. Los documentos fueron obtenidos de Common Crawl Foundation (2021). En esta sección se detalla cómo se obtuvieron los datos, filtrado, normalizado y preparación para generar los *word embeddings*.

Common Crawl ofrece varios conjuntos de documentos, usualmente mensuales. En esta investigación se escogió la semana 51 del año 2018 y la semana 25 del 2021. Se optó por escoger datos anteriores al 2019 para evitar posibles influencias de síntomas de COVID-19 en los datos. Y la semana 25 se escogió por ser la más reciente al momento de iniciar esta investigación. Cabe destacar que estas fechas se refieren a la fecha de recolección de los documentos, no a la fecha de los documentos. Estas colecciones son acumulativas y los documentos nuevos son minoría, por lo que los impactos estacionales (por ejemplo: fiestas de fin de año) deberían ser poco significativos.

5.1.2. Descarga de los Datos

Para descargar los datos se elaboró un programa en Go. Este recibe como argumento la fecha que se desea descargar. Posteriormente, el programa busca la lista de archivos que forman parte del corpus de dicha fecha. A partir de esa lista, el programa descarga dos conjuntos diferentes de archivos WARC: los metadatos y el texto plano. Los metadatos son necesarios pues contienen información sobre el idioma detectado y la codificación de cada documento. Una vez que se descargan los metadatos se seleccionan únicamente los documentos que tienen texto en español. Para evitar almacenar la totalidad de los datos, unos 32 TB por corpus (incluyendo metadatos y texto plano), el programa realiza el filtrado al mismo tiempo que se filtran los archivos documentos.

Los documentos extraídos del archivo WARC son procesados de la siguiente forma:

1. Se eliminan documentos que no estén en español.
2. Se convierte el documento a UTF-8.
3. Se eliminan caracteres que no sean letras latinas.
4. Se convierte el texto a minúscula.
5. Se eliminan documentos que no tengan un largo mínimo (50 caracteres).
6. Se separan los documentos en párrafos.
7. Se eliminan párrafos duplicados.
8. Se guarda el corpus en formato comprimido.

Para realizar todos los pasos anteriores de forma simultánea se utilizó la biblioteca *Datachan*[1], la cual fue elaborada por el autor para facilitar el procesamiento de datos de forma local siguiendo un modelo de programación similar a *MapReduce* (Dean y Ghemawat, 2004) o *Spark* (Zaharia, Chowdhury, Franklin, Shenker, y Stoica, 2010). En *Datachan* cada etapa puede comenzar a procesar datos tan pronto comienzan a ser emitidos por la etapa anterior, así que no necesita esperarse a que termine una etapa para comenzar la siguiente. Cada etapa se describe en más detalle en las siguientes secciones.

[1] `https://github.com/estebarb/datachan`

5.1.3. Detección de idioma

Para detectar el idioma se utilizaron los metadatos disponibles en CommonCrawl, donde la organización añadió los idiomas y su porcentaje por documento, según fueron detectados por el algoritmo CLD2[2]. Se descartaron los documentos que contuvieran menos de un 40 % de texto español.

La detección de idioma puede ser un proceso muy costoso en CPU. Inicialmente, se probaron varias bibliotecas en Go que detectan el idioma utilizando heurísticas. Sin embargo, los datos de perfilado mostraron que el programa estaba pasando la mayor parte del tiempo detectando el idioma y convirtiendo a UTF-8 (que requiere identificar primero la codificación original). Debido a esto se prefirió descargar los metadatos y utilizar los valores identificados por CommonCrawl, aunque eso significara duplicar el tamaño de los datos descargados.

5.1.4. Conversión a UTF-8

El texto fue convertido a UTF-8 utilizando la codificación detectada por CommonCrawl y que fue almacenada en los metadatos. Si la codificación detectada no era soportada por la biblioteca usada para realizar la conversión, se procedió a descartar el documento.

Sin embargo, una gran cantidad de texto en internet no está codificado correctamente. Por ejemplo, varios sitios web en UTF-8 contenían texto donde una cadena en UTF-8 había sido malinterpretada como ISO-8859-1 y se había vuelto a convertir a UTF-8. Se incluyó código adicional para corregir este error, en caso de que el texto contuviera los bytes en hexadecimal `C3 83`.

La detección de la codificación es un proceso que requiere mucho CPU, pues es necesario interpretar el texto según varias codificaciones y utilizar heurísticas para determinar si la codificación podría ser correcta o no. Inicialmente, se intentó determinar la codificación dentro del mismo programa que descargaba el corpus, pero debido al alto uso de CPU se procedió a utilizar los archivos de metadatos.

5.1.5. Normalización

Para normalizar el texto se eliminaron todos los caracteres que no fueran letras latinas utilizando una expresión regular. Luego se convirtió todo el texto a minúscula.

[2]https://github.com/CLD2Owners/cld2

Se consideraron hacer normalizaciones adicionales. Por ejemplo, limitar las palabras a un diccionario como Real Academia Española (2021). Sin embargo, estas normalizaciones fueron removidas ya que eliminaban palabras válidas o que serían interesantes para el análisis (como nombres propios y notablemente, el término COVID). Debido a esto, se prefirió mantener las palabras originales, aunque esto significara mantener errores ortográficos o de codificación.

5.1.6. Eliminar duplicados

Para eliminar las oraciones o párrafos duplicados se utilizó un filtro Bloom (Bloom, 1970). Este fue configurado para tener 1280000000 elementos y una probabilidad de colisiones de 10^{-7}.

5.1.7. Compresión del corpus

La compresión del corpus resultante nos permite guardar el corpus en menos espacio de disco duro. Antes de descargar los datos completos de CommonCrawl se hicieron estimaciones usando un subconjunto pequeño de los datos y se concluyó que se requerirían entre 400 GB y 800 GB por corpus. Algunos de los retos que esto representa son altos costos de transferencia, costos de almacenamiento, bajo ancho de banda y cuotas de almacenamiento limitadas en los equipos disponibles. Para reducir el impacto de estos problemas se decidió comprimir el corpus resultante.

Existen muchos algoritmos de compresión de datos. Sin embargo, este trabajo requeriría procesar dicho corpus utilizando *word2vec* para generar los *word embeddings*. De las implementaciones disponibles de forma pública y probadas en este trabajo, solamente Gensim soporta entradas comprimidas, en este caso con GZip. Las demás, como el *word2vec* original (Mikolov, Sutskever, et al., 2013) o variaciones como *pWord2vec* (Ji et al., 2016) y *BlazingText* (Gupta y Khare, 2017) requieren texto plano como entrada.

Para lograr comprimir el corpus y al mismo tiempo poder seguir usando las implementaciones disponibles de *word2vec* se utilizó un esquema de compresión personalizado, que se detalla en el Apéndice A en la página 77. En primer lugar, se tomó la lista de palabras de CREA (Real Academia Española, 2021) ordenadas por frecuencia. Luego, cada palabra del corpus fue transformada así: si la palabra está en CREA, entonces se reemplaza por la posición de la palabra en CREA y se guarda en base 62 (dígitos y letras mayúsculas y minúsculas); si no está en CREA entonces se escribe un signo de admiración y se copia la palabra exactamente igual. Cada palabra se separó

con un espacio en blanco (ASCII 32) y cada párrafo u oración se separó con un salto de línea (ASCII 10). De esta forma, se logró un esquema de compresión donde el texto sigue siendo texto UTF-8 plano válido y además se redujo el tamaño del corpus aproximadamente a la mitad: por ejemplo, en el caso de 2021-25, el corpus comprimido mide 86,6GB y si se descomprime mediría 171,87GB. En el Listado A.1 en la página 77 se muestra un ejemplo de implementación de este algoritmo en Python. En la Tabla 5.1 se muestra un ejemplo de texto comprimido utilizando este método.

	Texto
Comprimido	aH !covid 16 oR 1DB Y 8 qV 3Eg 6 44S 1ra LJ e 2 Qz 1nX d 4 !covid
Descomprimido	medidas covid donde podrán consultar todos los acuerdos circulares y comunicados informativos relacionados con la emergencia sanitaria por el covid

Tabla 5.1: Ejemplo de un texto comprimido con el algoritmo utilizado

Como cada palabra simplemente se transforma en otra cadena, por lo general más corta, el archivo resultante puede seguir siendo procesado por herramientas existentes como `cat`, `grep`, `wc` o bien programas que cuenten las palabras, calculen TF-IDF o generen *word embeddings*, sin necesidad de cambiar su código fuente. Por ejemplo, en el caso de *BlazingText*, lo único que hubo que ajustar fue configurar el largo mínimo de las palabras en 1 y luego revertir los *embeddings* resultantes para que tuvieran la palabra original y no la comprimida.

5.1.8. Características de los Corpus

Se generaron dos corpus de textos en español, a partir de los corpus recolectados por CommonCrawl de 2018 (semana 51) y 2021 (semana 25). Las características de dichos corpus se resumen en la Tabla 5.2.

	2018-51	2021-25
Total de documentos descargados	3 086 millones	2 394 millones
Documentos en español	131 millones	106 millones
Documentos con largo mínimo	131 millones	106 millones
Total de párrafos/oraciones	2 424 millones	2 157 millones
Total quitando oraciones repetidas	872 millones	756 millones
Total de palabras	35 480 millones	29 533 millones
Tamaño comprimido (base 62)	102,1GB	86,6GB

Tabla 5.2: Características de cada etapa de los corpus recolectados

5.1.9. Construcción de los *word embeddings*

La construcción de los *word embeddings* se realizó usando *BlazingText* (Gupta y Khare, 2017). Este es ofrecido como un servicio en la nube por Amazon Web Services. *BlazingText* usa las técnicas presentadas en Ji et al. (2016) para acelerar el cálculo de los *embeddings* usando el algoritmo *word2vec* o *fastText*, pero usando GPU en lugar de CPU.

La selección de *BlazingText* obedeció a un criterio de costo de procesamiento, rendimiento, soporte de las técnicas requeridas y que esté activamente soportado. Se analizaron diversas implementaciones para construir los modelos, dentro de las cuales destacan: *gensim* (Rehurek y Sojka, 2011), *pWord2Vec* (Ji et al., 2016) y *BlazingText* (Gupta y Khare, 2017). Un resumen de los aspectos considerados se encuentra en la Tabla 5.3[3].

Implementación	Palabras/segundo	Costo (5 épocas)
gensim	1M/s (1)	n/a
pWord2Vec	20M/s (2)	$0,068265USD/1M palabras
BlazingText	17M/s (3)	$0,249542USD/1M palabras

Tabla 5.3: Estimaciones de costo y rendimiento de diversas implementaciones de word embeddings

Dados los datos del Cuadro 5.3 lo más económico era utilizar *pWord2Vec*. Ji et al. (2016) publicaron el código fuente en GitHub, pero este estaba optimizado para procesadores Xeon Phi. Debido a que no se tenía acceso a un clúster con procesadores Xeon Phi se optó por migrar el código para que funcionara en procesadores ARM. La conversión fue exitosa y funcionó correctamente al ser probada en instancias c6gd.16xlarge de AWS (ARM Graviton 2, 64 núcleos y 128 GB de RAM). Sin embargo, el programa tuvo varios fallos al ser ejecutado sobre el corpus completo (en su mayoría desbordamientos aritméticos por usar variables de 32 bits). Debido a esto, se prefirió utilizar otra implementación, para evitar el riesgo de introducir errores que pudieran afectar la generación de los *embeddings* al intentar arreglar el código. La versión de *pWord2vec* migrada a ARM está disponible en `https://github.com/estebarb/pWord2Vec`.

[3]Respecto a la Tabla 5.3:
(1) En un AMD Ryzen 7 5800X de 8 núcleos con SMT. Luego de 4 núcleos no escala linealmente.
(2) En un c6gd.16xlarge: ARM Graviton 2 con 64 núcleos, 128 GB de RAM y $2,4576USD/hora. Escala linealmente por cada núcleo. Precios a noviembre del 2021.
(3) Resultado de ejecución propia. 8 instancias ml.c4.8xlarge de 36 núcleos, 60 GB de RAM y $1,909 USD por hora cada instancia. Precios a noviembre del 2021.

La ejecución de *BlazingText* se hizo en 8 instancias *ml.c4.8xlarge* de AWS. Cada una de estas instancias tiene 36 núcleos, 60 GB de RAM y cuestan $1,909USD/hora. El corpus del 2021 tardó 15852 segundos en ejecutarse, mientras que el corpus del 2018 tardó 18639 segundos. El costo total de entrenar ambos modelos fue de $150,07USD. La configuración usada se muestra en la Tabla 5.4.

batch size	12	buckets	10 000 000
early stopping	no	epochs	5
learning rate	0,5	max char	35
min char	1	min count	10
min epochs	2	mode	batch skipgram
negative samples	5	patience	4
sampling threshold	0,0001	subwords	false
vector dimension	300	window size	5
word ngram	2		

Tabla 5.4: Configuración usada para *BlazingText*

5.1.10. Alineamiento de los *word embeddings*

Dado que la construcción de los *word embeddings* tiene aspectos aleatorios, no es posible garantizar que la misma palabra tendrá vectores comparables en diferentes modelos (Hamilton et al., 2016). Para corregir esto es necesario alinear los vectores.

Existen varios algoritmos para alinear vectores y según la tarea que se esté realizando algunos pueden ser mejores que otros. Por ejemplo, Joulin et al. (2017) es usado para alinear *embeddings* entre diferentes idiomas de forma no supervisada. En este trabajo se escogió usar el mismo método usado por Hamilton et al. (2016), que es tratar los vectores como en el problema de Procrustes ortogonales. La solución a este problema consiste en encontrar la rotación de la matriz que minimice la distancia entre los pares de vectores (cada palabra en el modelo del año A y la misma palabra en el modelo del año B), la cual se puede encontrar eficientemente usando descomposición en valores singulares (Schönemann, 1966).

En la Figura 5.1 en la página siguiente se muestra un histograma de las diferencias de coseno entre los *word embeddings* de las palabras en los modelos entrenados con datos del 2018 y 2021, luego del proceso de alineamiento. Acá el eje horizontal corresponde a la similitud de coseno, donde 1 quiere decir que el vector es idéntico y -1 que es totalmente opuesto. El eje vertical corresponde al total de palabras con dicha similitud de coseno. El gráfico muestra una forma de similar a una media campana con el máximo cerca de 1, lo cual coincide con la expectativa de que la mayoría

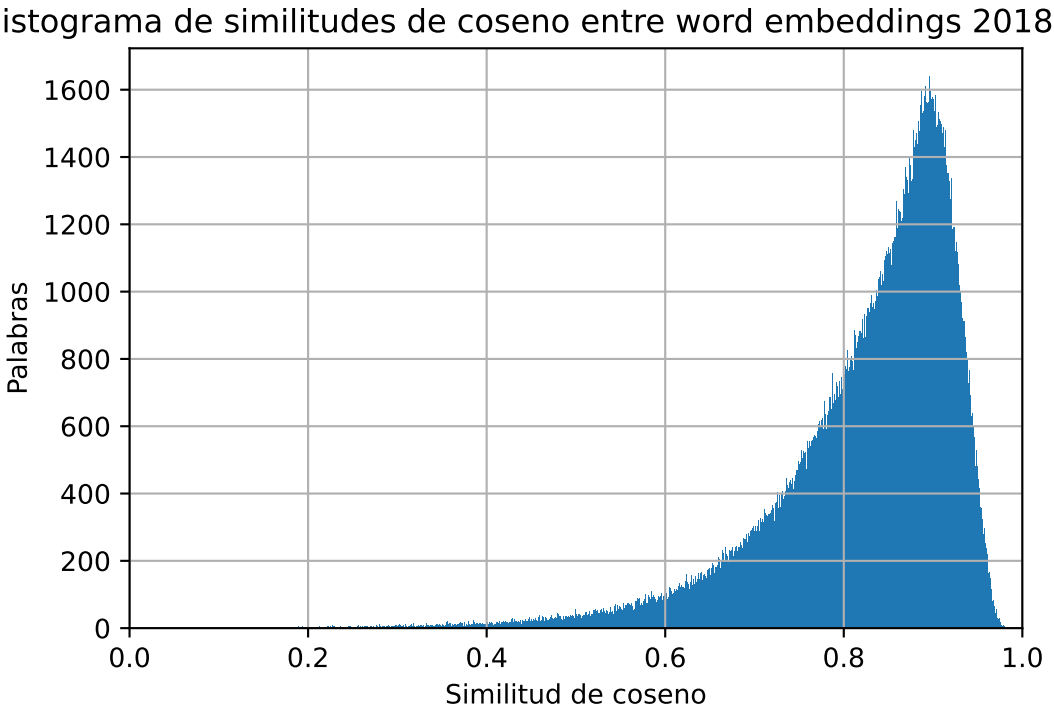

Figura 5.1: Histograma de la similitud de coseno entre las palabras en los *word embeddings* de 2018 y 2021, luego del proceso de alineamiento, con 1000 intervalos

de las palabras tienen un desplazamiento semántico muy pequeño. El motivo por el cual no es una media campana «perfecta» es por la construcción de los *word embeddings*, que introducen cierta aleatorización y además los corpus de entrenamiento son diferentes. Sin embargo, este comportamiento es esperado.

5.2. Resultados de Desplazamiento Semántico

Para detectar las palabras con más desplazamiento semántico se tomaron las palabras con una similitud de coseno menor o igual a 0,7 (entre 2018-51 y 2021-25) y una frecuencia en 2021-25 mayor a 1000 ocurrencias. De las 265505 palabras en común entre ambos modelos y más de 1000 ocurrencias se identificaron 24943 con una similitud de coseno inferior o igual a 0,7.

Al aplicar un algoritmo de agrupamiento sobre estas palabras se encontraron un total de 575 clústeres. Los clústeres fueron ordenados según la menor similitud de coseno presente en las palabras del clúster de forma ascendente. Luego de revisar los clústeres manualmente se y descartar clústeres compuestos mayoritariamente por errores de codificación, errores ortográficos, código fuente o términos sin sentido, se identificaron 35 temas. De estos, se seleccionaron los siguientes tres temas

relacionados con el COVID:

1. COVID: El clúster contiene sinónimos de COVID como coronavirus y términos asociados como pandemia o confinamientos. No incluye vacunas.

2. Mascarillas: Incluye sinónimos de mascarillas como cubrebocas, barbijo o tapaboca.

3. Vacunación: Incluye sinónimos de vacunarse, como inoculará o inmunizar.

5.2.1. Palabras con errores ortográficos o de codificación

Tal como se mencionó en la Apartado 5.1.5 en la página 29 sobre Normalización, en esta investigación se evitó eliminar o modificar las palabras, para no perder términos que podrían ser interesantes (como nombres o COVID). Sin embargo, esta decisión causó que algunos términos con errores ortográficos o con errores de codificación se mantuvieran en el estudio.

En los siguientes listados de palabras hay términos con errores ortográficos o de codificación. Estos errores provienen del corpus, que es el objeto de estudio, y no son errores introducidos al elaborar este documento. Algunos ejemplos de estos errores son: «protecciã*f*æ» (posiblemente un error de codificación, donde la palabra original era «protección»), «recesiã» (un error de codificación donde no se nota cual era la palabra original), «aztrazeneca» (un error ortográfico común al referirse a la empresa «AstraZeneca») o «máscarilla» (un error ortográfico, ya que la palabra no lleva tilde).

Para prevenir modificaciones accidentales a las palabras estudiadas (por ejemplo, accidentalmente corregir un error ortográfico) se automatizó la generación de los gráficos, tablas y listados de palabras (listas de palabras en el clúster, palabras más cercanas, palabras más lejanas, palabras que se acercaron más y palabras que se alejaron más) a partir de los datos.

5.2.2. COVID-19

El clúster de covid incluye palabras relacionadas con el COVID-19. El centroide de este clúster es la palabra *pandemia*. En promedio, la similitud de coseno de las palabras en el clúster, respecto a sí mismas en 2018 y 2021 es de 0,4802.

Se muestran cuatro figuras que resumen los desplazamientos alrededor del clúster covid. En la Figura 5.2 en la página 37 se muestran los vecinos más cercanos en

el 2018. En la Figura 5.3 en la página 38 se muestran los vecinos más cercanos en el 2021. La Figura 5.4 en la página 39 muestra las palabras que se acercaron más a términos dentro del clúster, mientras que la Figura 5.5 en la página 40 muestra las palabras que se alejaron más. En estas cuatro figuras, las palabras dentro del clúster se muestran en negro, el centróide del clúster en rojo y las palabras de cada caso (vecinos o mayores desplazamientos) en azul.

Palabras en clúster: covd, pandemia, pandémico, covit, desconfinamiento, covi, coronavirus, covid, cuarentena, pandémica, cov, pandémicas, pandemía, cuarentenas, confinamiento, confinamientos, autoaislamiento, sars, pandémicos, ncov, pandemiaâ y contagiados

Vecinos más cercanos en 2018: epidemia, contagiadas, sosprechado, iclr, pospandémico, infodemia, optm, epidémicas, itors, gripecita, vacunaría, autoconfinamiento, wpac, grupoesoc, swers, protecciãfæ, sanibrun, thinkbook, puntosfuertes, mubs, epidémicos, adenovirus, ftui, stalkerware y epidémico

Vecinos más cercanos en 2021: contagiadas, desescalada, contagios, infectados, crisisâ, fallecidos, virus, contagiado, influenza, fallecimientos, epidemia, confirmados, hospitalizados, actualâ, patógeno, crisis, norovirus, recesiã, rebrotes, contagiaron, pandã, micaâ, contagio, hospitalizadas y asintomáticos

Palabras que se acercaron más: desescalada, virus, contagios, aspo, influenza, fallecidos, crisisâ, contagio, rebrotes, wuhan, contagiadas, patógeno, fallecimientos, convid, antiviral, ébola, epidemia, hantavirus, decesos, brote, distanciamiento, dengue, sanitaria, norovirus y contagiarse

Palabras que se alejaron más: ufsa, sanibrun, generalmills, maxdisplay, amdpress, nnum, mubs, dezombies, tnkr, simblicamente, fundamentall, bpage, gozazaragoza, websties, jsst, wpac, thinkbook, agroconcept, binor, puntosfuertes, swers, itors, optm, sosprechado y iclr

Aunque el COVID-19 era una enfermedad totalmente desconocida antes de finales del 2019, muchos de los términos en este clúster eran de uso común con anterioridad. Debido a esto, algunos términos de uso común se acercaron a otros términos relacionados con la pandemia de COVID-19. Una excepción es el término «COVID», que no era de uso común antes de la aparición de esta enfermedad: en este caso dicha palabra antes estaba asociada a otras palabras sin sentido.

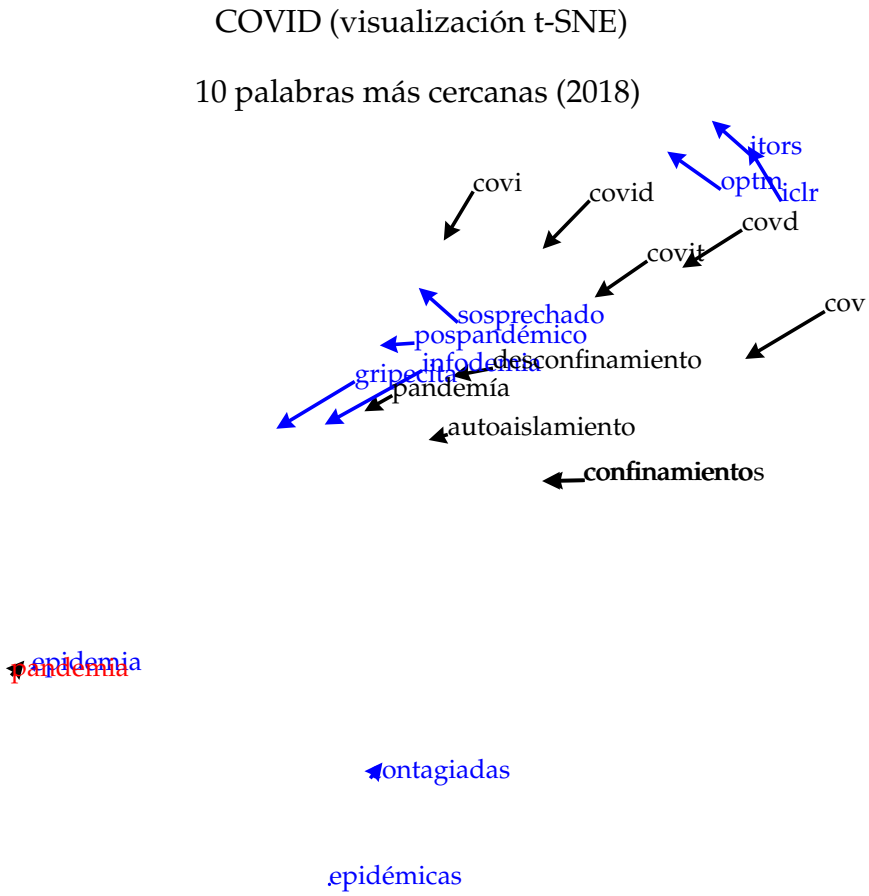

Figura 5.2: Análisis de vecinos más cercanos en el 2018 al clúster sobre COVID-19

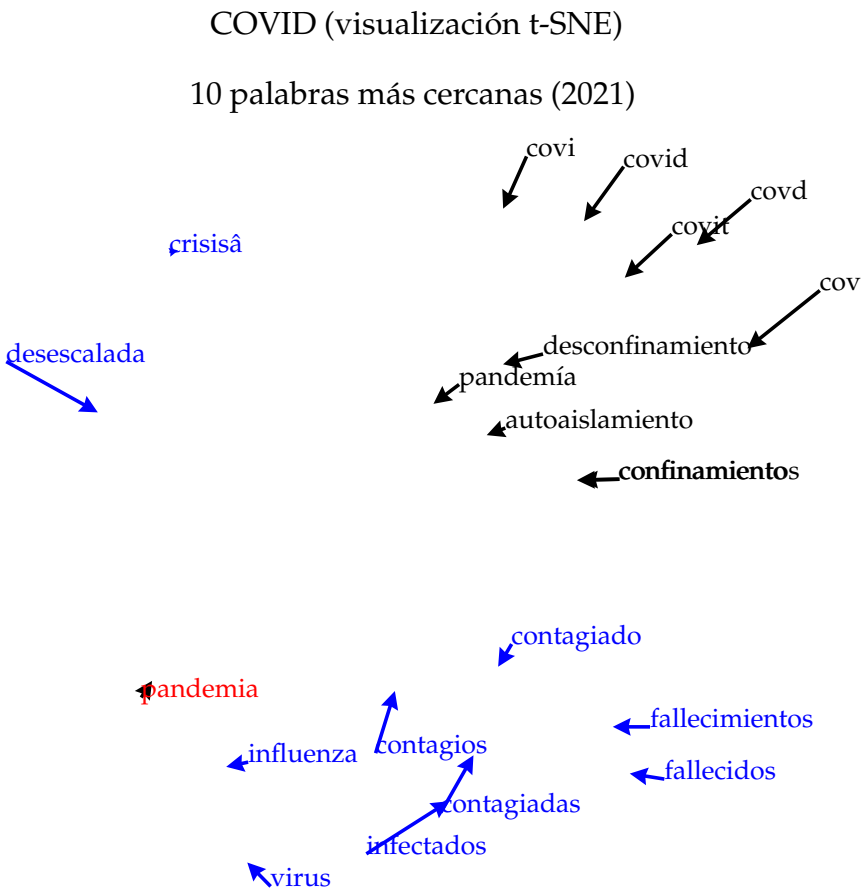

Figura 5.3: Análisis de vecinos más cercanos en el 2018 al clúster sobre COVID-19

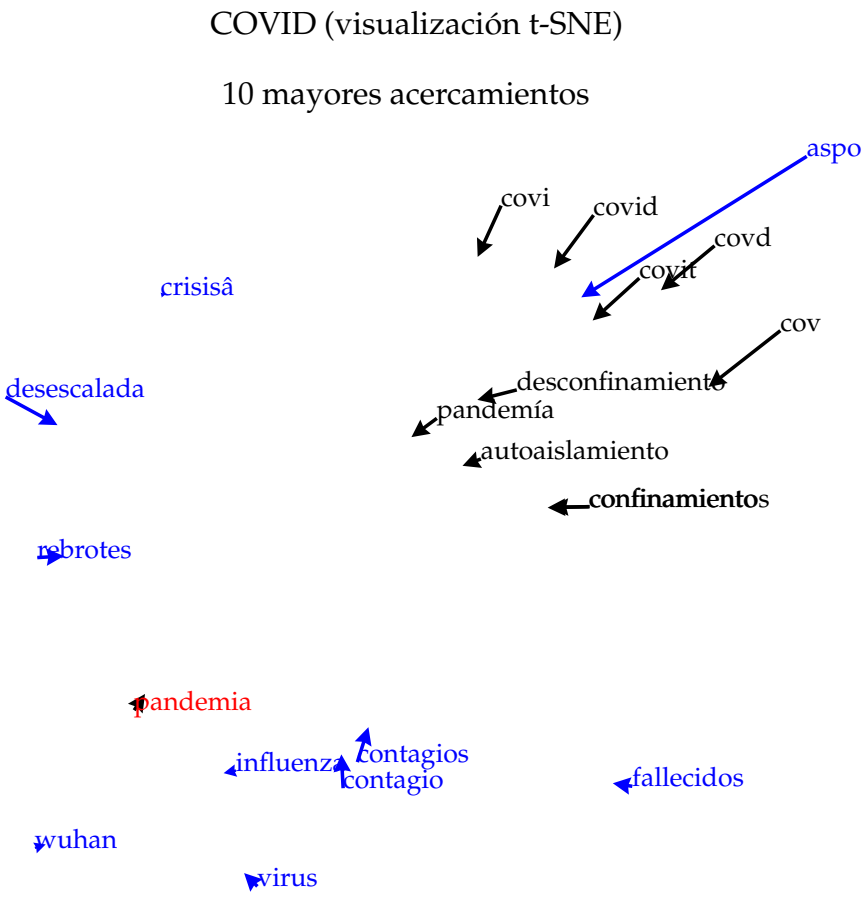

Figura 5.4: Análisis de mayores acercamientos al clúster sobre COVID-19

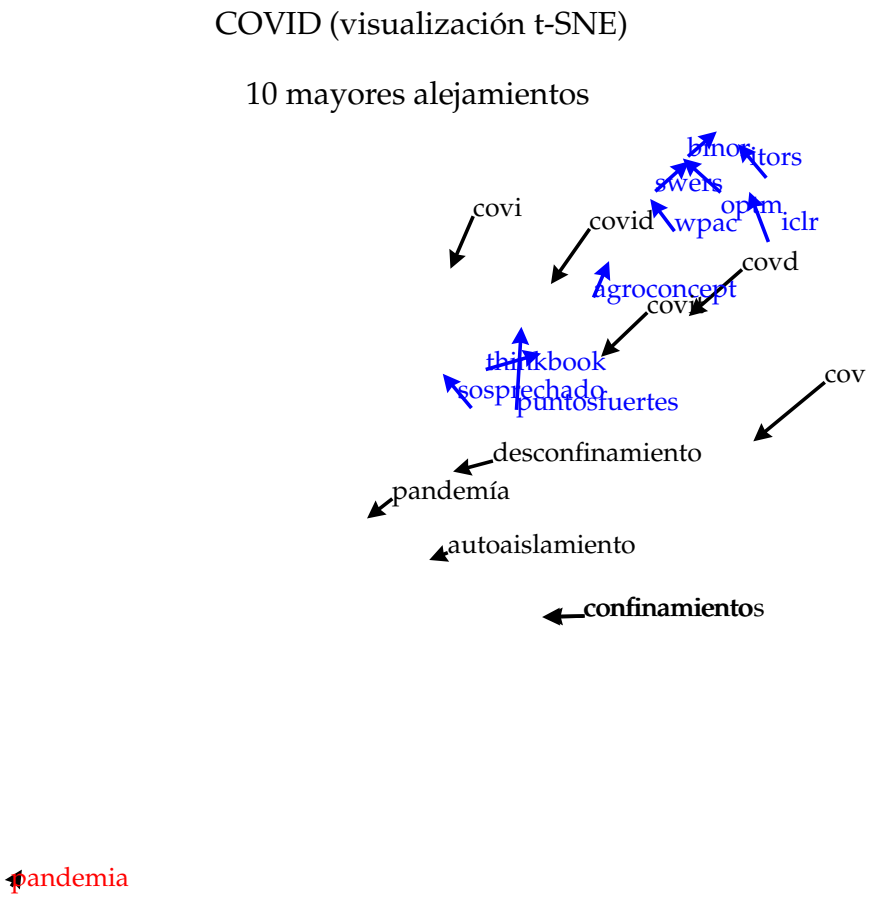

Figura 5.5: Análisis de mayores alejamientos al clúster sobre COVID-19

El término «COVID»

En el caso de «covid», este término anteriormente estaba asociado a términos que parecen errores ortográficos o de codificación, como por ejemplo *amdpress* o *gozazaragoza*. Sin embargo, en el 2021 se asocia con términos para referirse al COVID-19 (incluyendo errores ortográficos) como covit, convid, covd. También se asocia a contagio, contagios, virus, pandemia, coronavirus, influenza y contagiados. En la Figura 5.6 se muestran los vecinos del término «covid» en el 2018 y en la Figura 5.7 en la página siguiente se muestran los vecinos en el 2021. En la Figura 5.8 en la página 43 están las palabras que más se acercaron al término «covid», mientras que en Figura 5.9 en la página 44 se muestran las que más se alejaron.

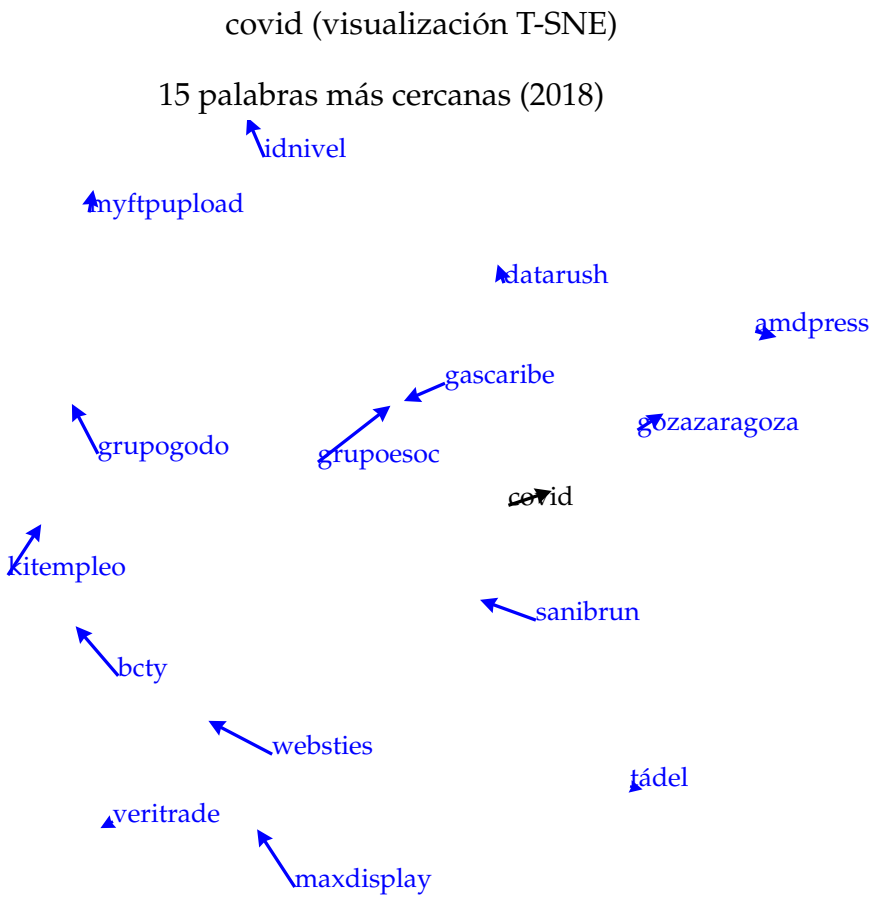

Figura 5.6: Vecinos más cercanos en el 2018 al término «COVID»

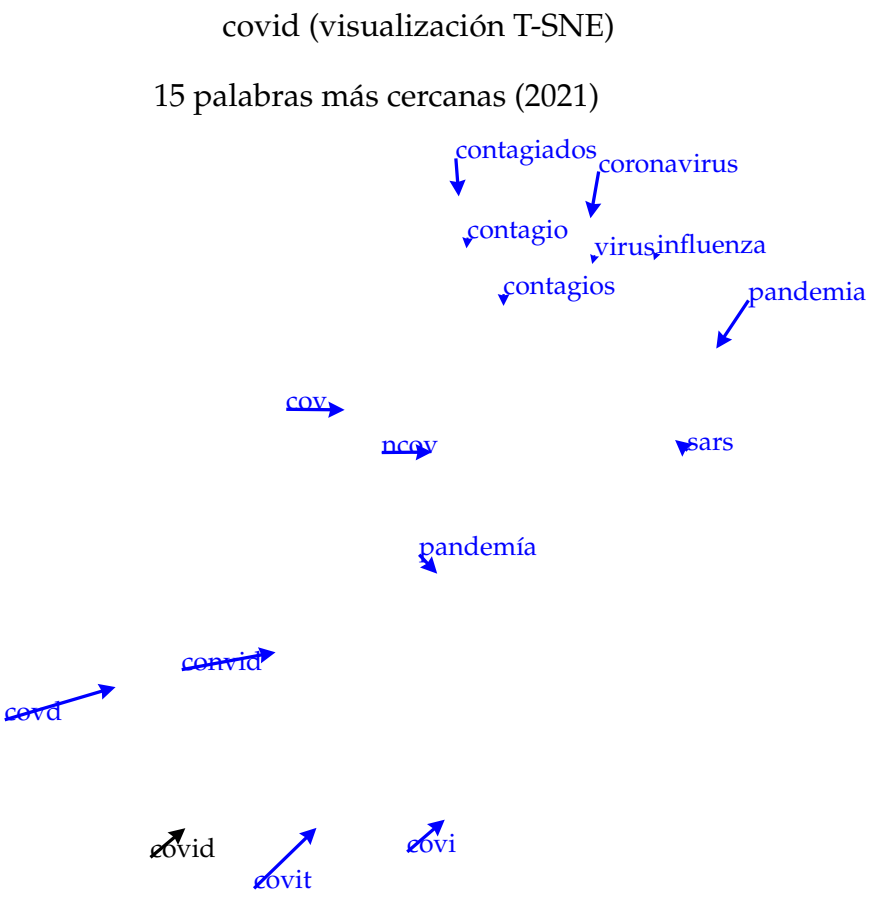

Figura 5.7: Vecinos más cercanos en el 2021 al término «COVID»

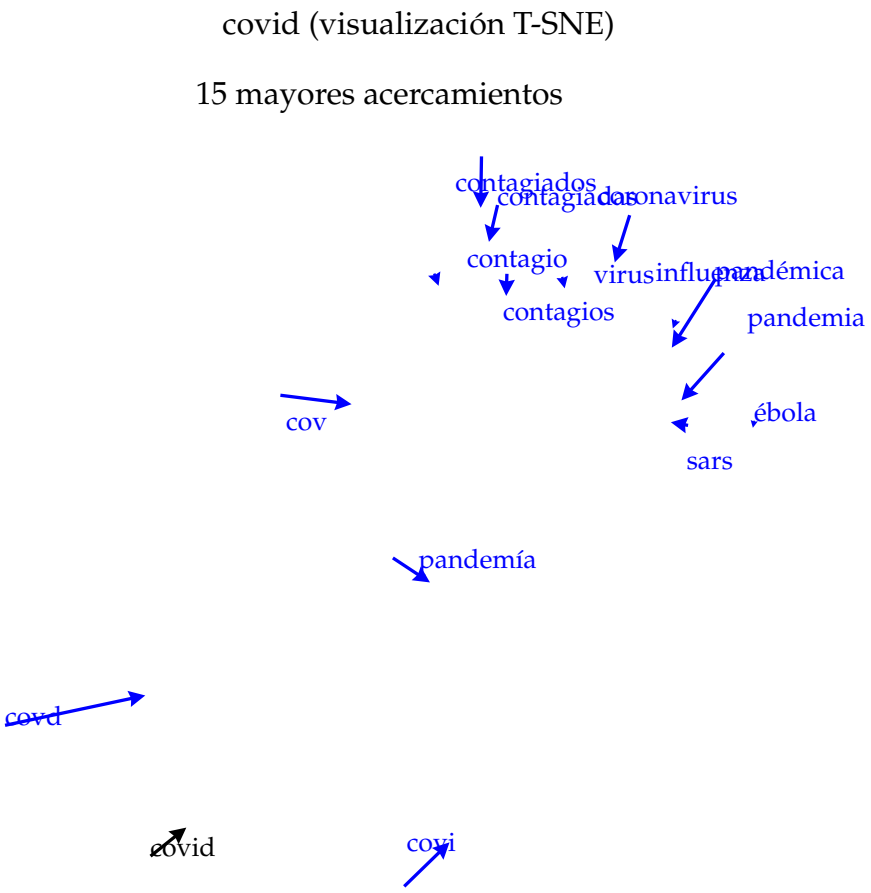

Figura 5.8: Palabras que se acercaron más al término «COVID»

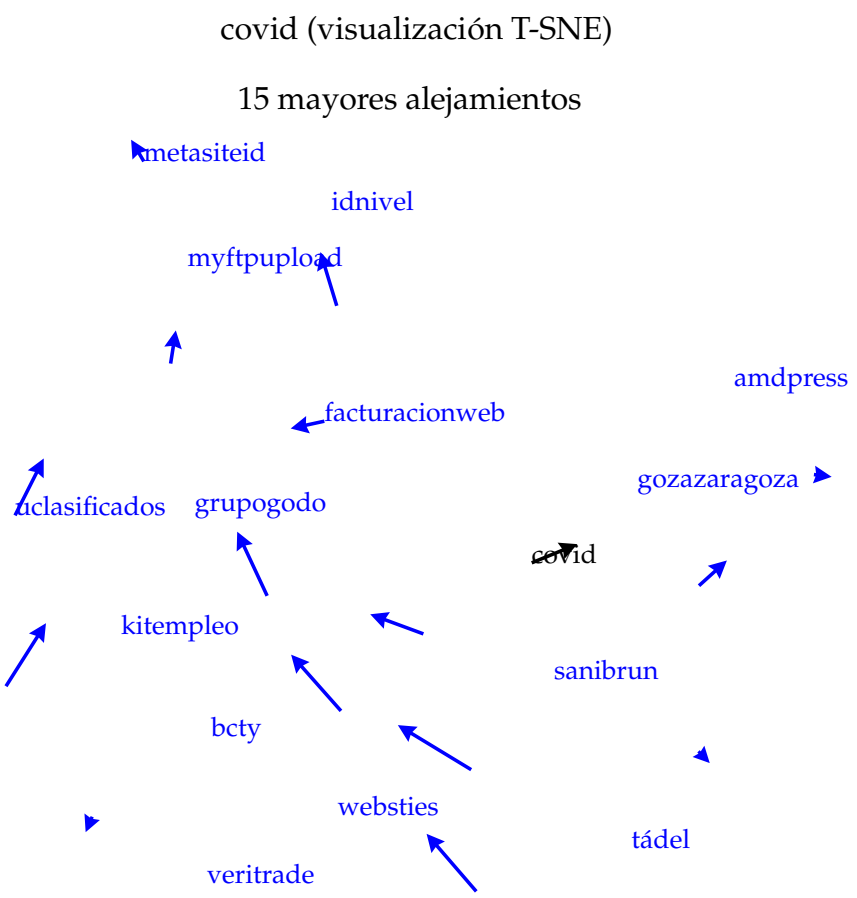

Figura 5.9: Palabras que se alejaron más al término «COVID»

El término «coronavirus»

El término «coronavirus» anteriormente estaba asociado a otros tipos de virus, como adenovirus, enterovirus, rotavirus, parvovirus, herpesvirus, parainfluenza, sincitial, rinovirus, flavivirus y arbovirus. También estaba cercano a bacterias como bordetella y bugdorferi. Sin embargo, estas cercanías cambiaron abruptamente y de hecho varias de las palabras más cercanas en 2018 fueron las que más se alejaron. En la Figura 5.10 se muestran los vecinos del término «coronavirus» en el 2018 y en la Figura 5.11 en la página siguiente se muestran los vecinos en el 2021. En la Figura 5.12 en la página 47 están las palabras que más se acercaron al término «coronavirus», mientras que en Figura 5.13 en la página 48 se muestran las que más se alejaron.

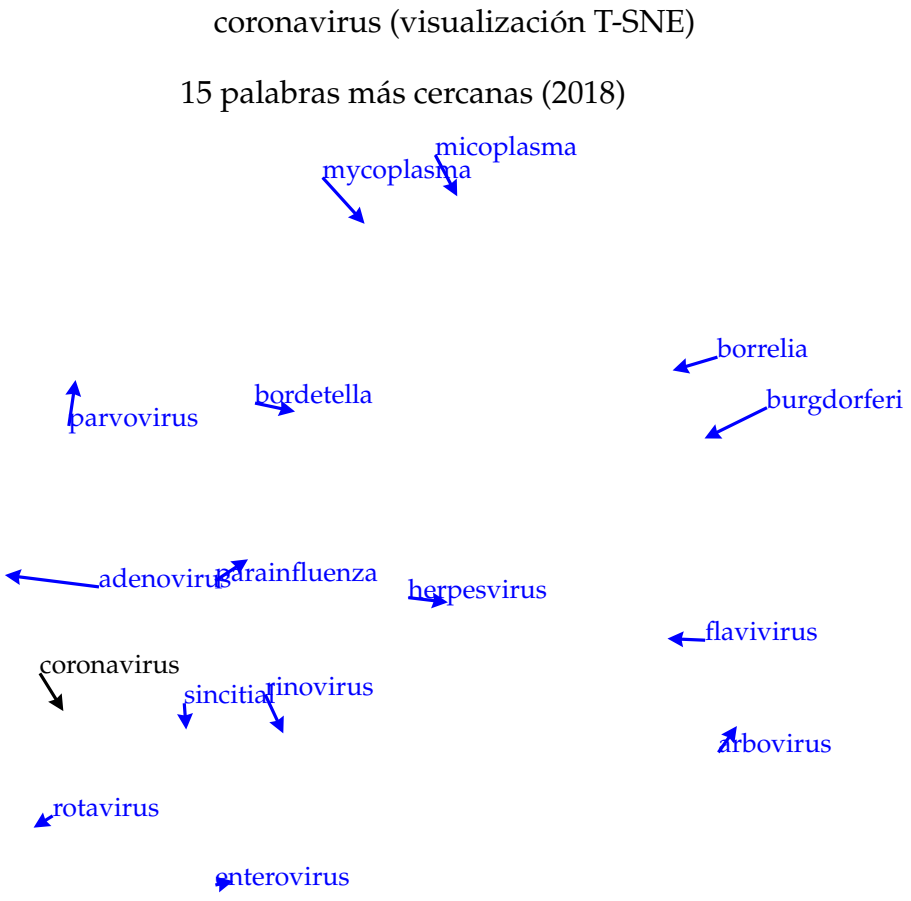

Figura 5.10: Vecinos más cercanos en el 2018 al término «coronavirus»

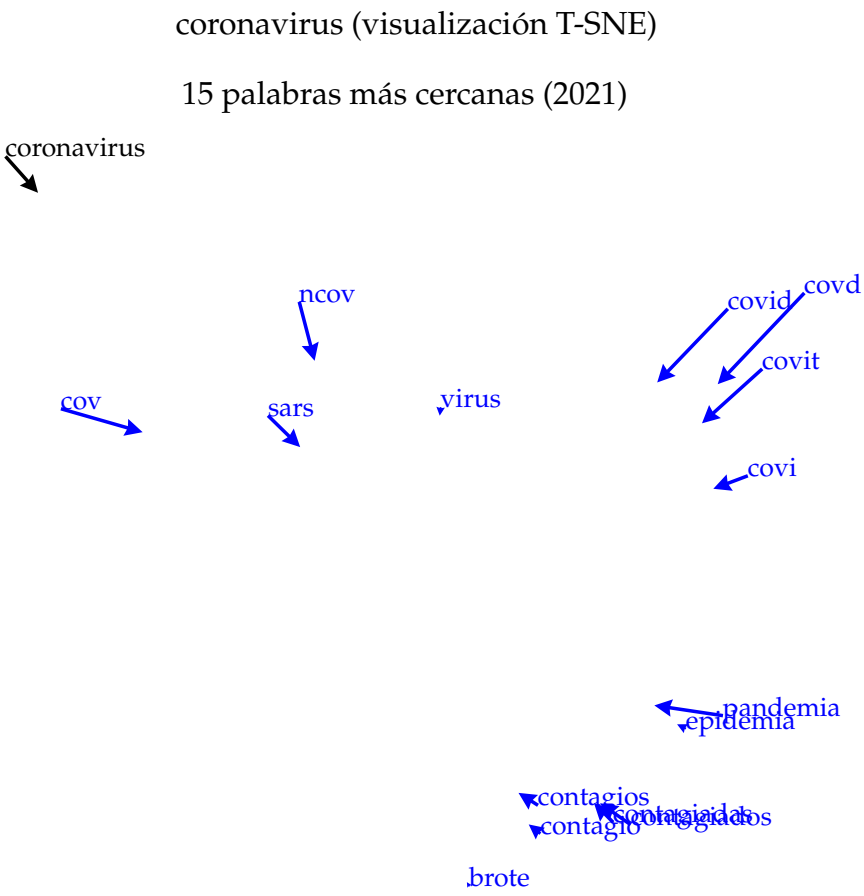

Figura 5.11: Vecinos más cercanos en el 2021 al término «coronavirus»

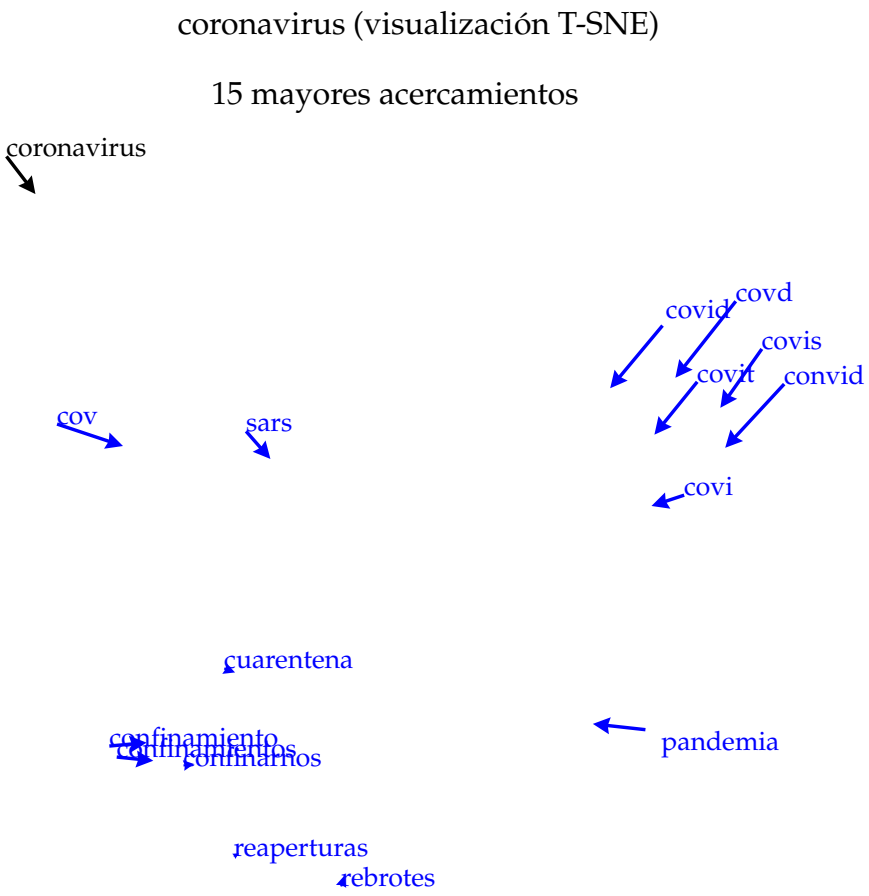

Figura 5.12: Palabras que se acercaron más al término «coronavirus»

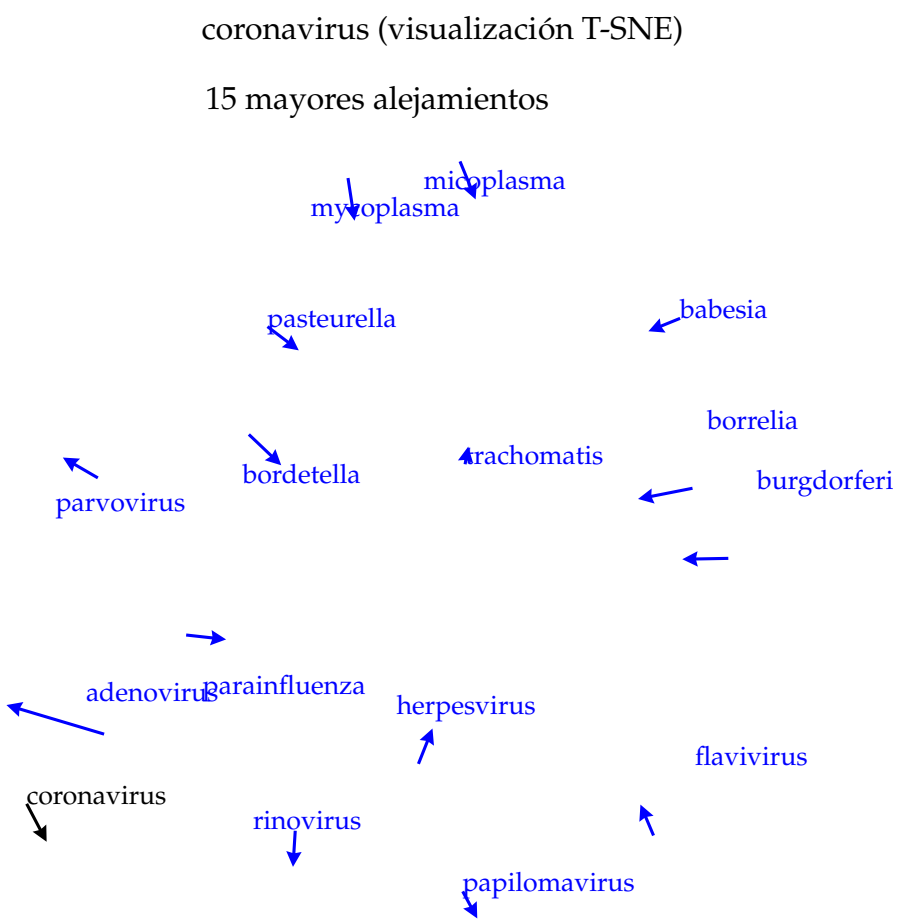

Figura 5.13: Palabras que se alejaron más al término «coronavirus»

El término «cuarentena»

El término «cuarentena» dejó de ser cercano a términos relacionados con «zoosanitario» o salmonela y se acercó a términos más afines a la pandemia del COVID-19, como coronavirus, pandemia o covid. También se acercó a términos relacionados con la nueva cotidianeidad, como teletrabajo, presencialidad o autoaislamiento. En la Figura 5.14 se muestran los vecinos del término «cuarentena» en el 2018 y en la Figura 5.15 en la página siguiente se muestran los vecinos en el 2021. En la Figura 5.16 en la página 51 están las palabras que más se acercaron al término «cuarentena», mientras que en Figura 5.17 en la página 52 se muestran las que más se alejaron.

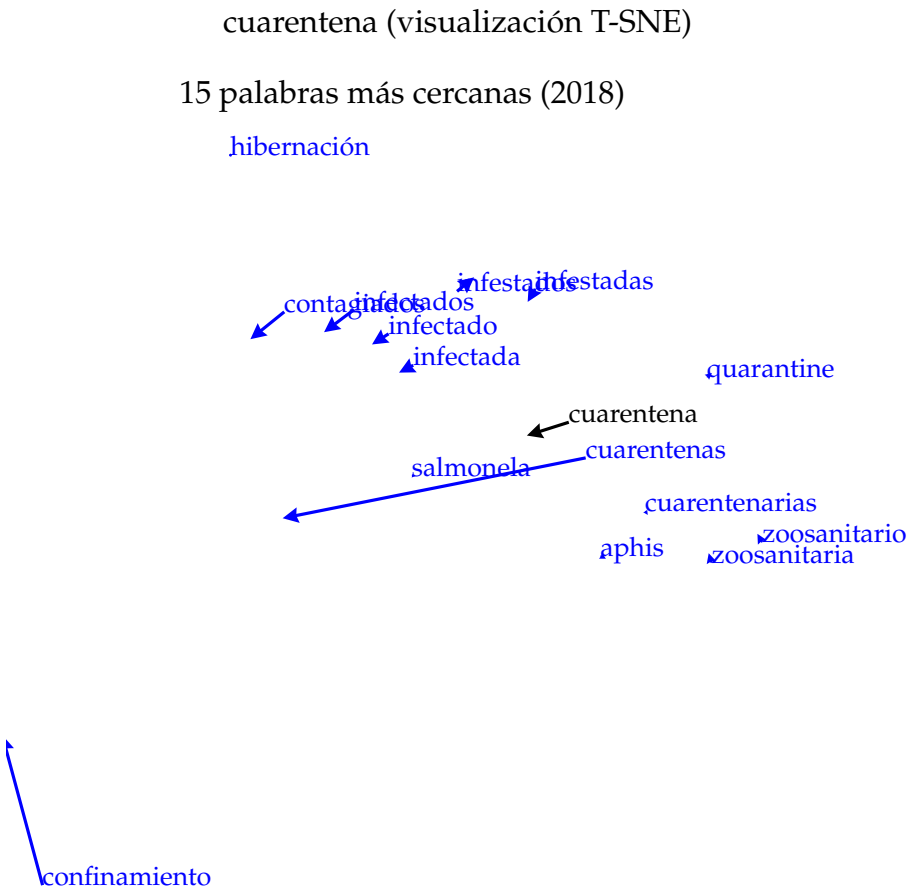

Figura 5.14: Vecinos más cercanos en el 2018 al término «cuarentena»

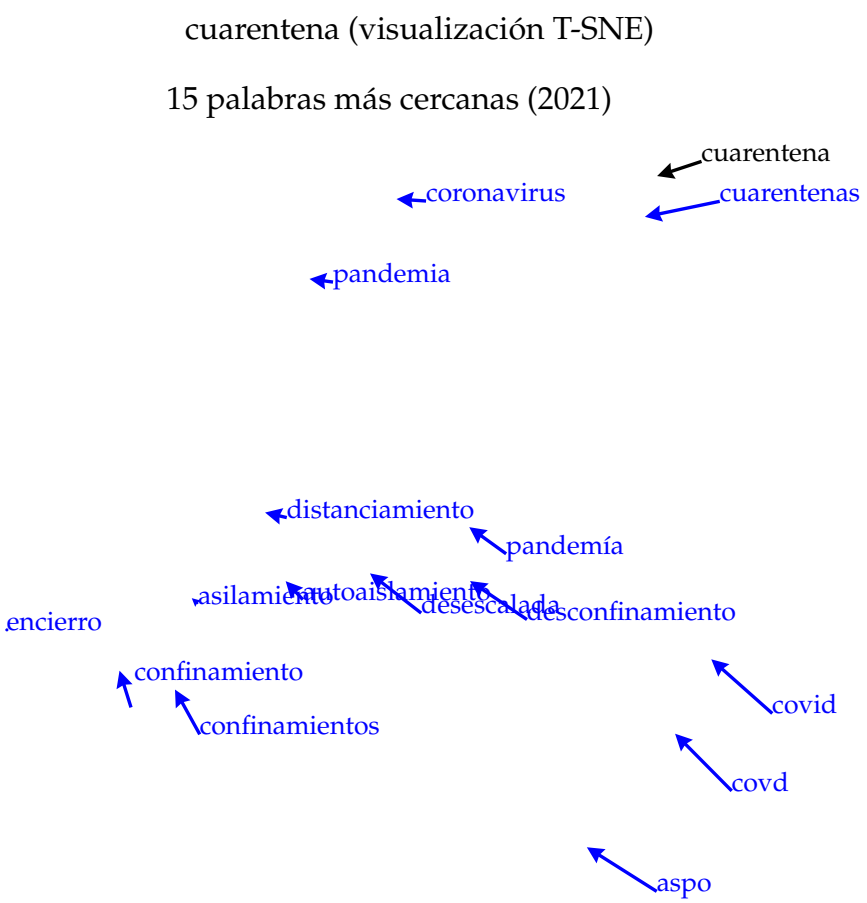

Figura 5.15: Vecinos más cercanos en el 2021 al término «cuarentena»

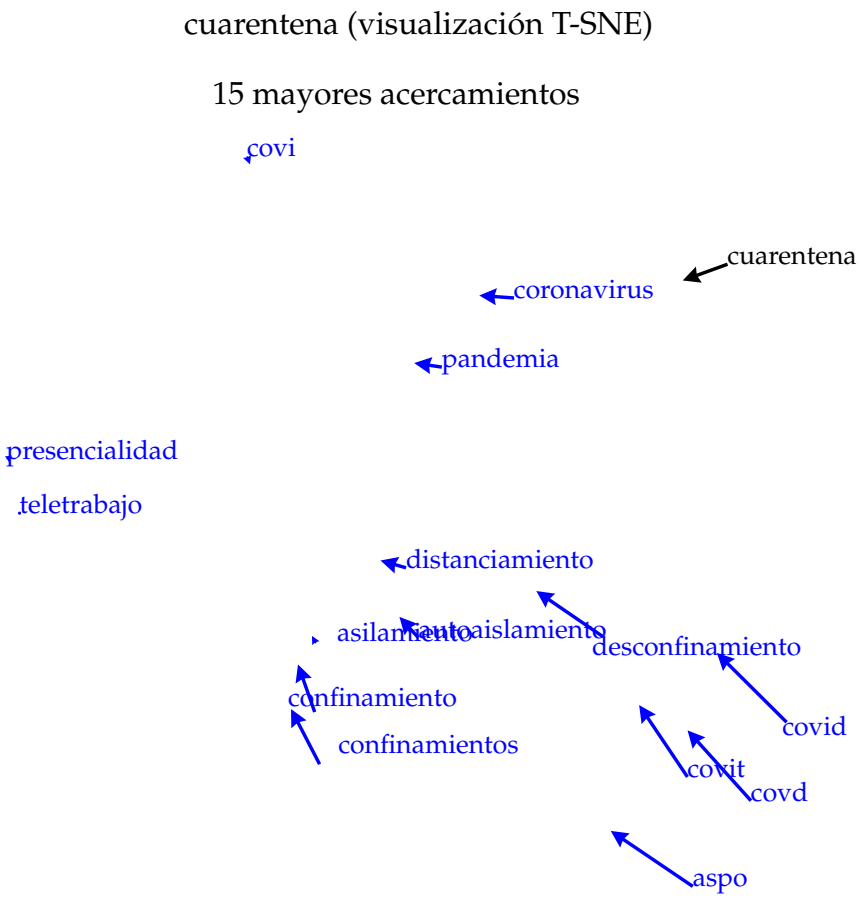

Figura 5.16: Palabras que se acercaron más al término «cuarentena»

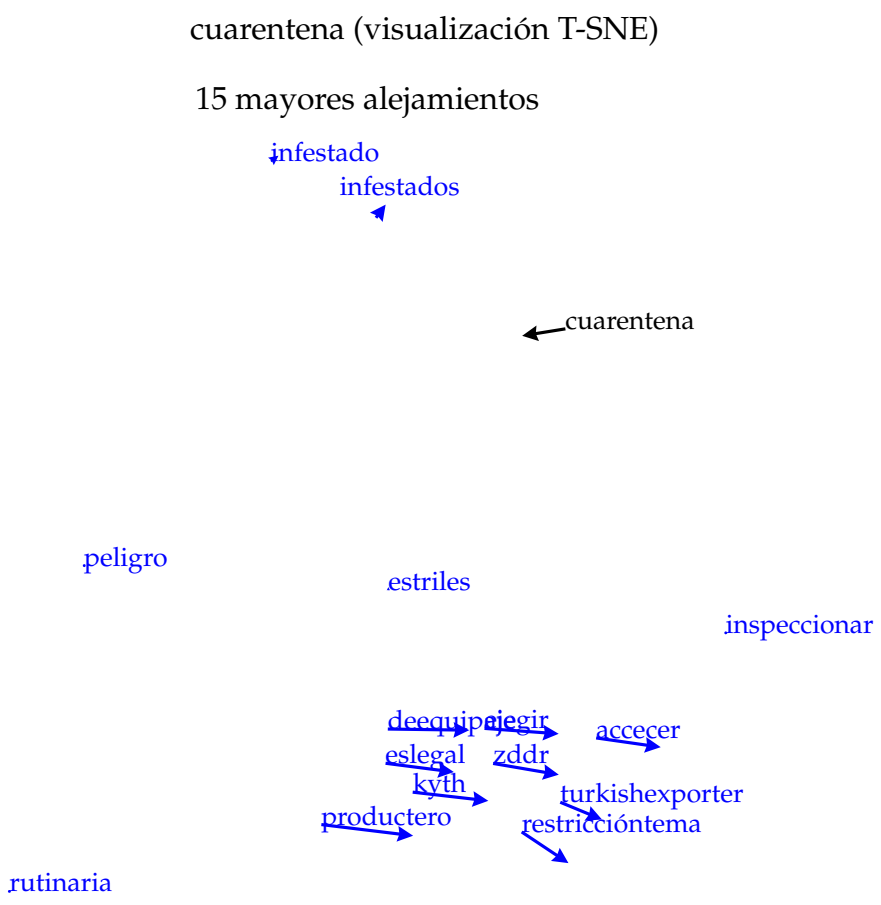

Figura 5.17: Palabras que se alejaron más al término «cuarentena»

El término «pandemia»

Anteriormente, el término «pandemia» estaba cercano a otros como epidemia, aviaria, influenza o gripe. Ahora, el término tiene como vecinos más cercanos a términos como coronavirus, covid, confinamiento o cuarentena. Dentro de los términos que más se alejaron de «pandemia» está bioterrorismo, antivacuna, gripe, aviar, aviaria, ébola, carbunco, tiroidea, poliomielitis y multirresistente. En la Figura 5.18 se muestran los vecinos del término «pandemia» en el 2018 y en la Figura 5.19 en la página siguiente se muestran los vecinos en el 2021. En la Figura 5.20 en la página 55 están las palabras que más se acercaron al término «pandemia», mientras que en Figura 5.21 en la página 56 se muestran las que más se alejaron.

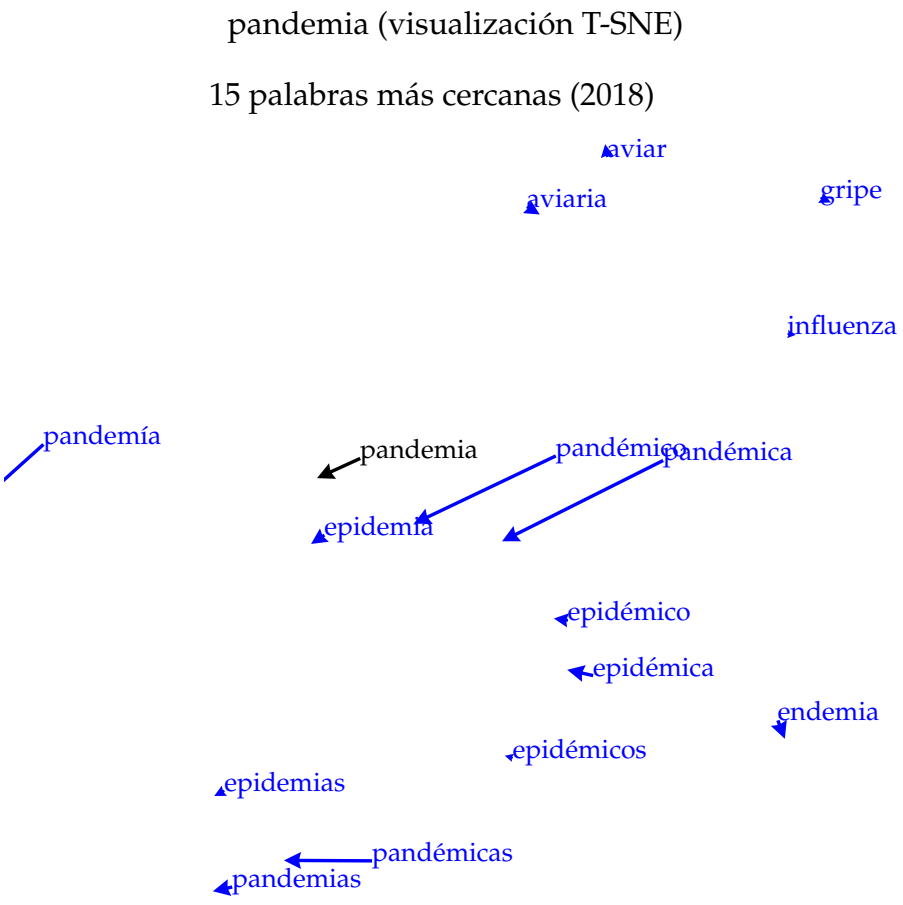

Figura 5.18: Vecinos más cercanos en el 2018 al término «pandemia»

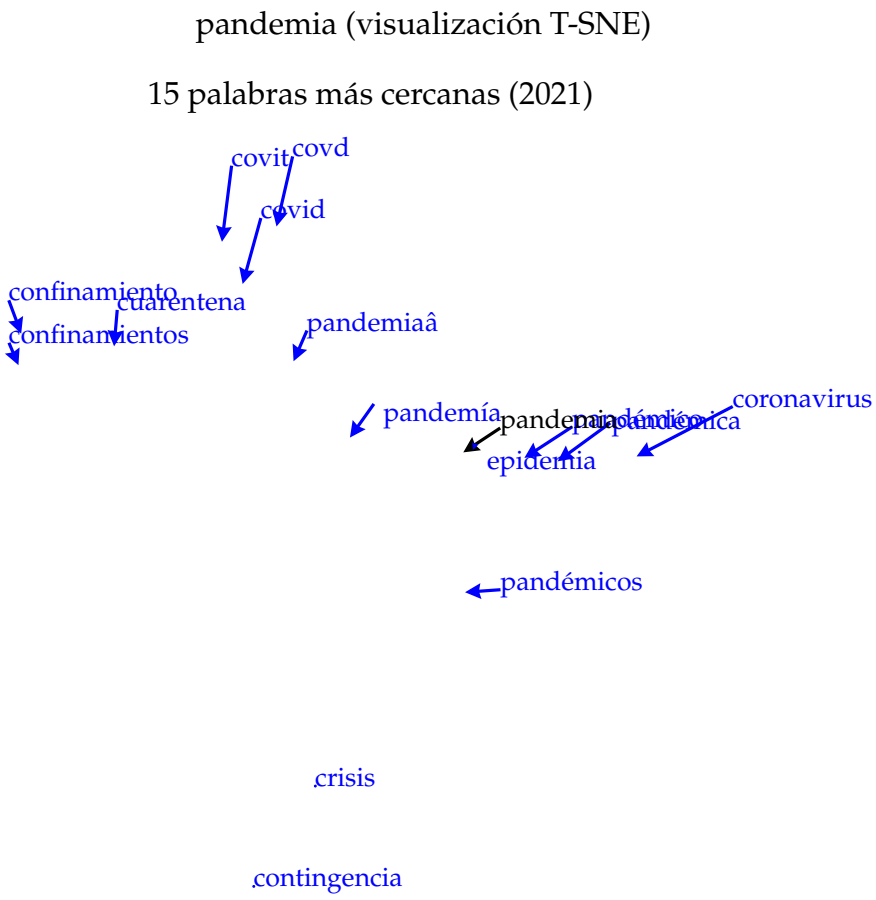

Figura 5.19: Vecinos más cercanos en el 2021 al término «pandemia»

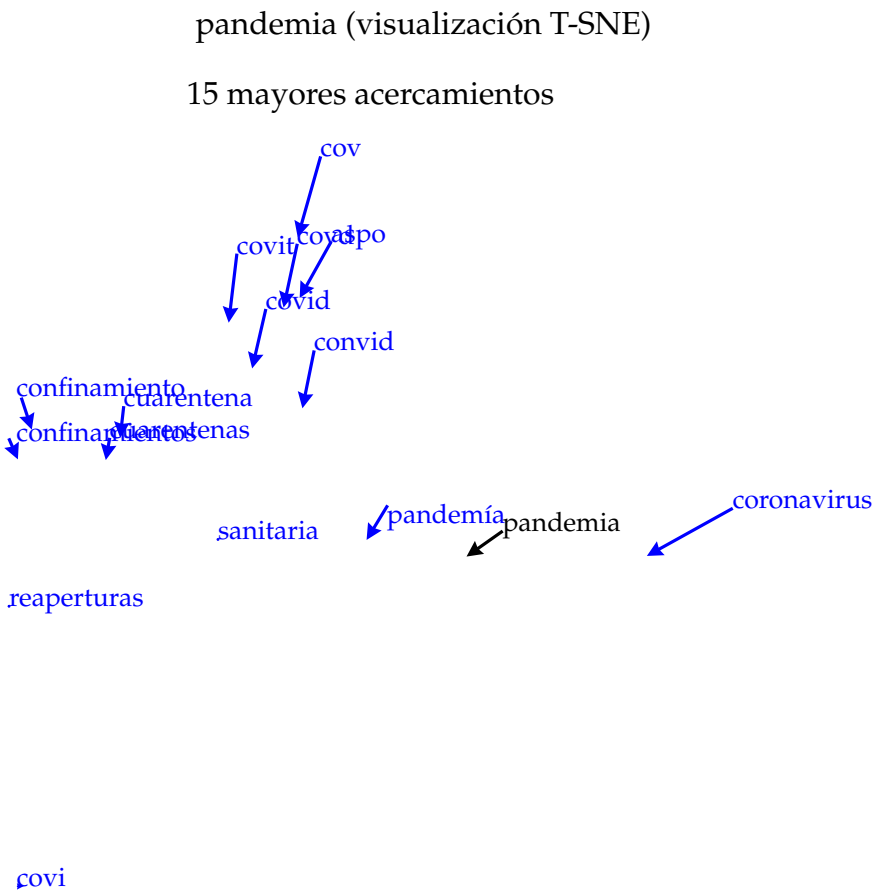

Figura 5.20: Palabras que se acercaron más al término «pandemia»

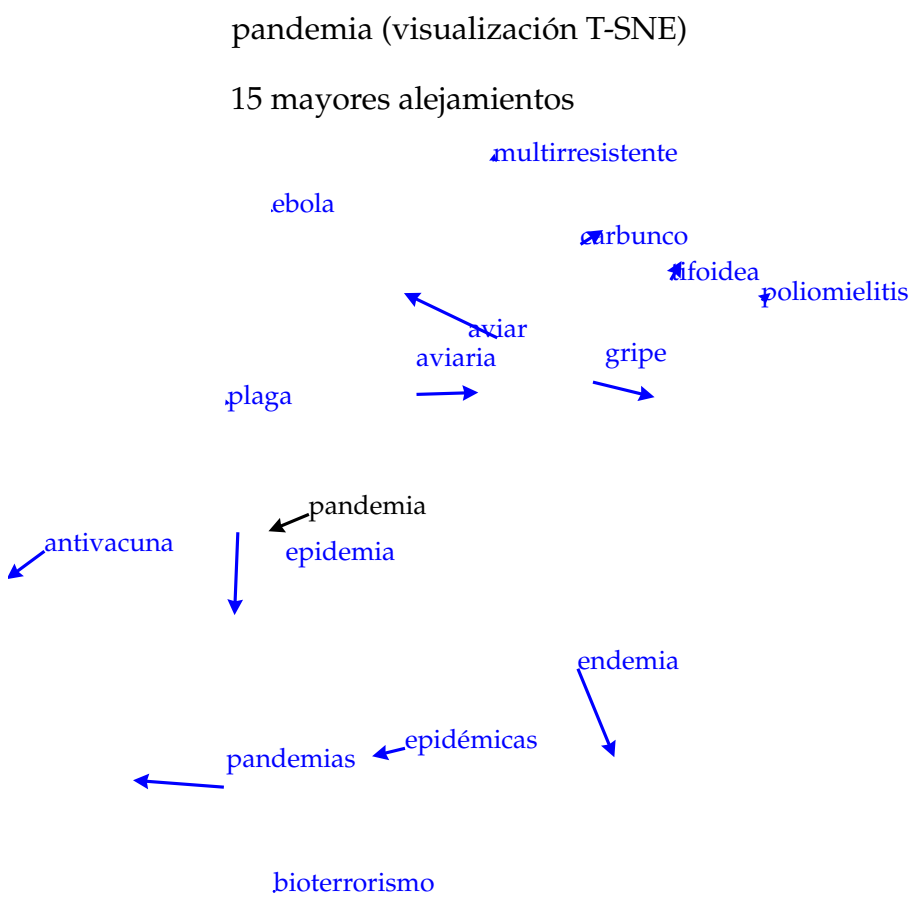

Figura 5.21: Palabras que se alejaron más al término «pandemia»

5.2.3. Mascarillas

El clúster de mascarillas incluye palabras relacionadas con mascarillas y protección facial. El centroide de este clúster es la palabra *cubrebocas*. En promedio, la similitud de coseno de las palabras en el clúster, respecto a sí mismas en 2018 y 2021 es de 0,5654.

Se muestran cuatro figuras que resumen los desplazamientos alrededor del clúster mascarillas. En la Figura 5.22 en la página siguiente se muestran los vecinos más cercanos en el 2018. En la Figura 5.23 en la página 59 se muestran los vecinos más cercanos en el 2021. La Figura 5.24 en la página 60 muestra las palabras que se acercaron más a términos dentro del clúster, mientras que la Figura 5.25 en la página 61 muestra las palabras que se alejaron más. En estas cuatro figuras, las palabras dentro del clúster se muestran en negro, el centróide del clúster en rojo y las palabras de cada caso (vecinos o mayores desplazamientos) en azul.

Palabras en clúster: mascarilla, nasobucos, distanciamiento, ffp, mascarillas, barbijo, cubrebocas, máscarillas, tapaboca, nasobuco y cubreboca

Vecinos más cercanos en 2018: máscarilla, hidratante, tapabocas, peeloff, pantalonespantalones, zddr, protecciã*f*æ, sosprechado, sérum, exfoliante, daytox, cuticate, zddd, deequipaje, enalimentos, purificante, infomodule, exfoliantes, hisopar, dqsa, buscadon, emuaidmax, barbijos, loción y facialderm

Vecinos más cercanos en 2021: tapabocas, barbijos, máscarilla, autofiltrantes, autofiltrante, máscaras, máscara, antibacterial, distanciamientos, fpp, camisolines, autoaislamiento, hidroalcohólicos, earloop, iir, sanitizantes, sanitizante, higiénicas, alcoholado, contagio, hidroalcohólicas, aglomeraciones, respiradores, batas y cuarentena

Palabras que se acercaron más: tapabocas, bioseguridad, aforos, barbijos, fpp, iir, cuarentena, aglomeraciones, aforo, cuarentenas, alcoholado, hidroalcohólicos, máscaras, sanitización, sanitizantes, aspo, antibacterial, sanitarias, hidroalcohólico, quirúrgicas, cuidándonos, higienizarse, sanitizados, gatell y higiénicas

Palabras que se alejaron más: pajaadictos, yardwe, puntosfuertes, superlarge, stalkerware, multianuncios, ebtools, plkd, reluzcas, chicklets, howtopronounce, dezombies, thinkbook, pantalonespantalones, dqsa, enalimentos, meojores, faciltarte, zddd, buscadon, sosprechado, cuticate, protecciã*f*æ, zddr y infomodule

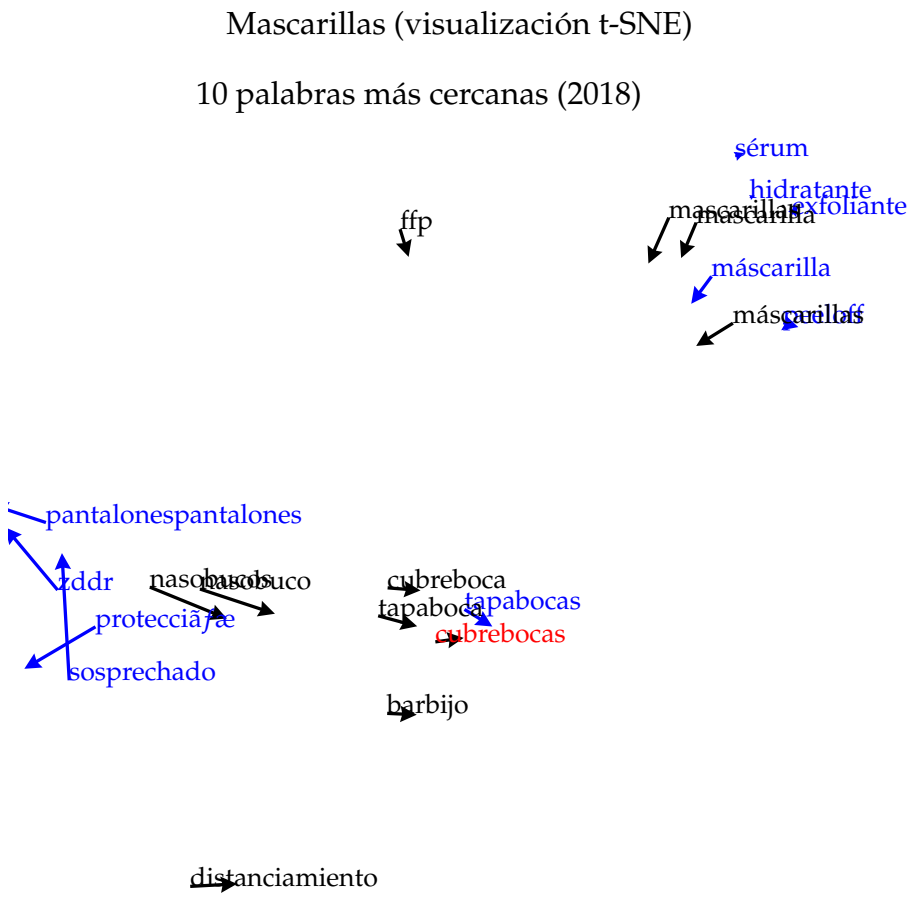

Figura 5.22: Análisis de vecinos más cercanos en el 2018 al clúster sobre mascarillas

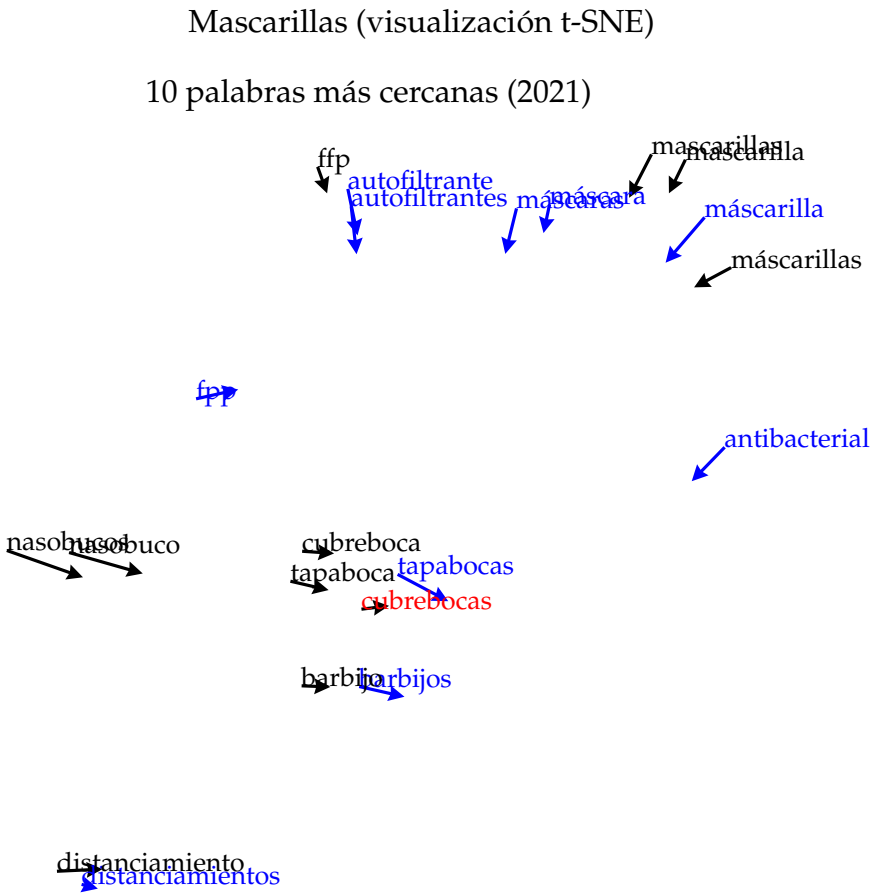

Figura 5.23: Análisis de vecinos más cercanos en el 2018 al clúster sobre mascarillas

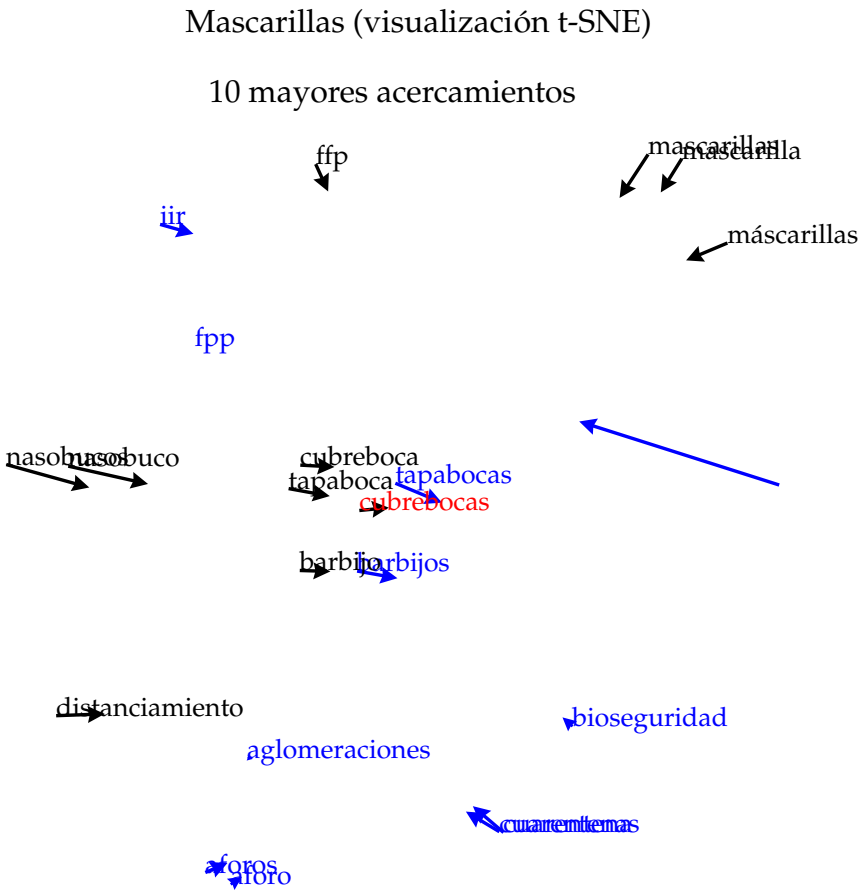

Figura 5.24: Análisis de mayores acercamientos al clúster sobre mascarillas

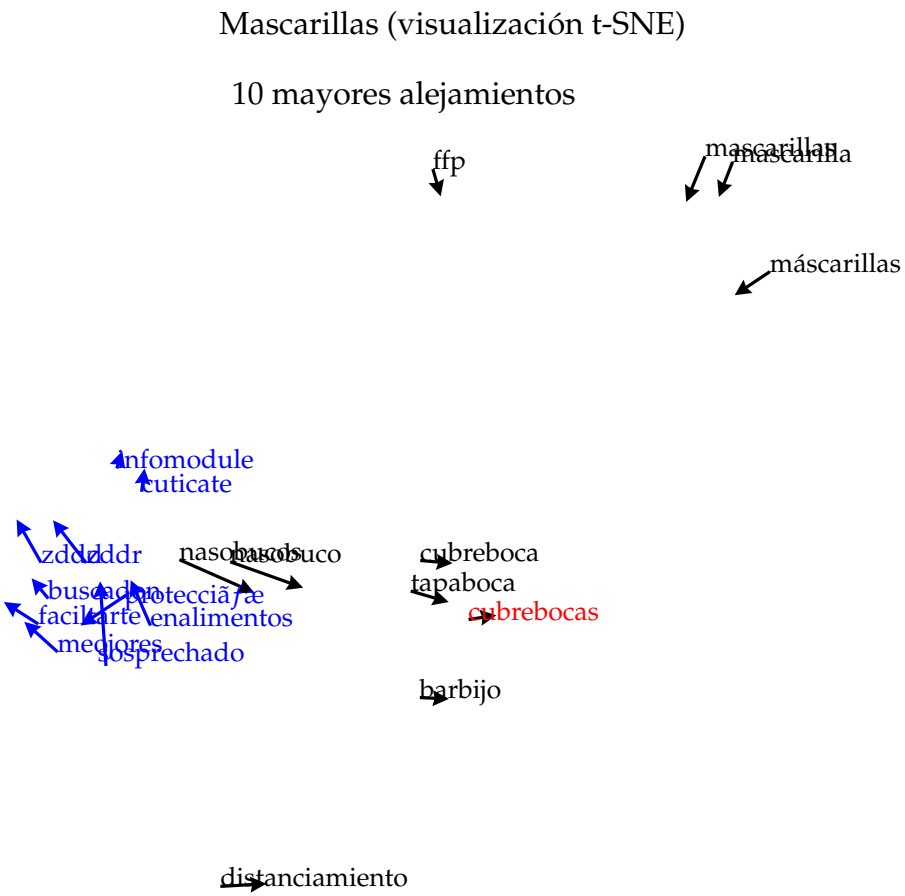

Figura 5.25: Análisis de mayores alejamientos al clúster sobre mascarillas

En el clúster de mascarillas se encuentran varios sinónimos de esta herramienta para proteger tanto la boca como la nariz. Es curiosa la presencia del término «nasobuco», ya que su uso se da principalmente en Cuba[4]. Es un caso interesante de cómo los *word embeddings* pueden capturar palabras con usos análogos en diferentes regiones del mundo. Otro aspecto interesante es que hubo un alejamiento respecto a las mascarillas faciales o tratamientos estéticos: términos como hidratante, peeloff, sérum, exfoliante, daytox, purificante, loción y facialderm que eran los vecinos más cercanos en el 2018 dejaron de serlo en el 2021.

5.2.4. Vacunación en general

El clúster de vacunación incluye palabras relacionadas con vacunación en general. El centroide de este clúster es la palabra *inmunizados*. En promedio, la similitud de coseno de las palabras en el clúster, respecto a sí mismas en 2018 y 2021 es de 0,6649.

Se muestran cuatro figuras que resumen los desplazamientos alrededor del clúster vacunación. En la Figura 5.26 en la página siguiente se muestran los vecinos más cercanos en el 2018. En la Figura 5.27 en la página 64 se muestran los vecinos más cercanos en el 2021. La Figura 5.28 en la página 65 muestra las palabras que se acercaron más a términos dentro del clúster, mientras que la Figura 5.29 en la página 66 muestra las palabras que se alejaron más. En estas cuatro figuras, las palabras dentro del clúster se muestran en negro, el centróide del clúster en rojo y las palabras de cada caso (vecinos o mayores desplazamientos) en azul.

Palabras en clúster: inoculadas, vacunarán, vacunada, inmunizada, inmunizado, inoculará, inoculados, inoculaciones, inmunizará, inmunizados, vacunará, inmunizando, inoculación, vacunó, vacunados, inmunizó y inocularse

Vecinos más cercanos en 2018: vacunadas, vacunado, vacunaron, desparasitada, inmunizaron, vacunándose, desparasitados, vacunan, inocularon, inoculada, inmunizadas, inoculado, desparasitado, inmunizarse, vacunando, inoculaciã, vacunar, vacunaran, vacunen, vacunaría, esterilizada, vacunación, vacunemos, pandemía y inocular

Vecinos más cercanos en 2021: inmunizarse, vacunarse, vacunando, vacunaron, vacunadas, inmunizadas, vacunado, inmunizaron, inmunización, vacunación, inoculado, vacunan, inoculada, inocular, inmunizar, inoculando, inocularon, va-

[4]Según `https://www.rae.es/observatorio-de-palabras/nasobuco`

cunar, vacunas, vacunaciones, inoculó, inoculacia̋, vacunara, vacunaría y inmunizaciones

Palabras que se acercaron más: vacunarse, inmunizarse, vacunado, sinopharm, vacunación, vacunando, inmunización, astrazeneca, biontech, inmunizadas, astrazéneca, covax, pfizer, cansino, sputnik, aztrazeneca, zeneca, inoculada, vacunadas, inmunizar, vacunaron, inoculado, covid, vacunar y alabí

Palabras que se alejaron más: infodemia, cuticate, dismuyen, youghourt, zoonóticas, pandémico, dqsa, variedadesde, sosprechado, tuberculosas, sonicación, lobosnews, nezuko, esterilizada, sindemia, pandémicas, vacunándose, hbtes, lisados, pospandémico, pandémicos, desparasitado, desparasitados, pandemía y desparasitada

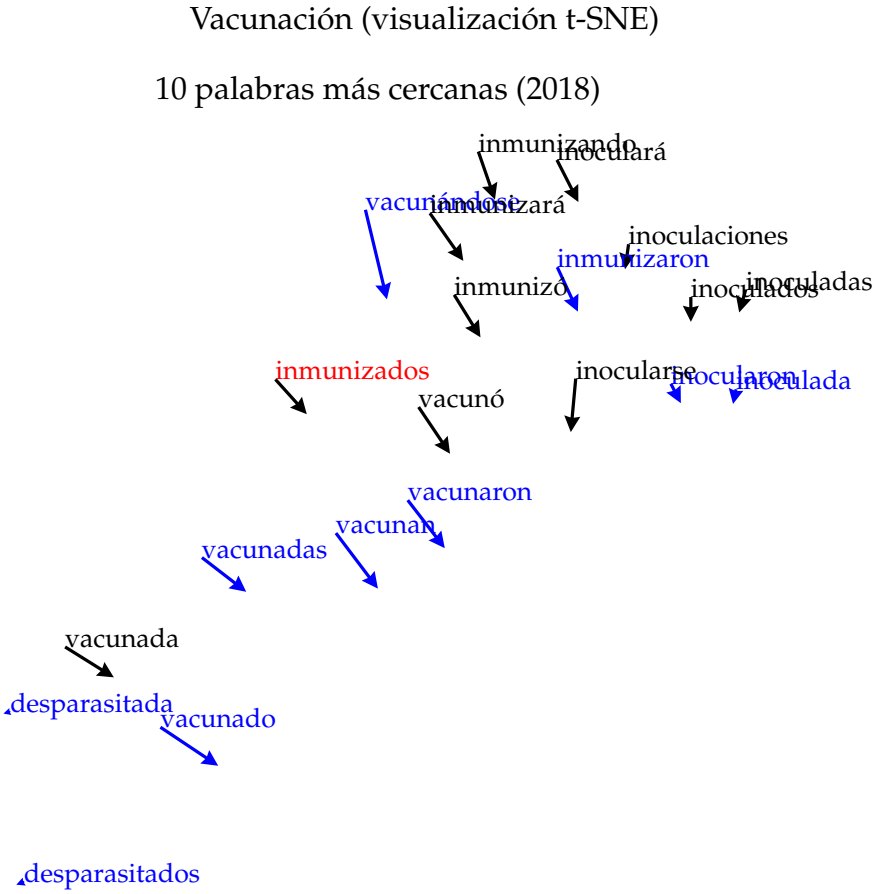

Figura 5.26: Análisis de vecinos más cercanos en el 2018 al clúster sobre vacunacion

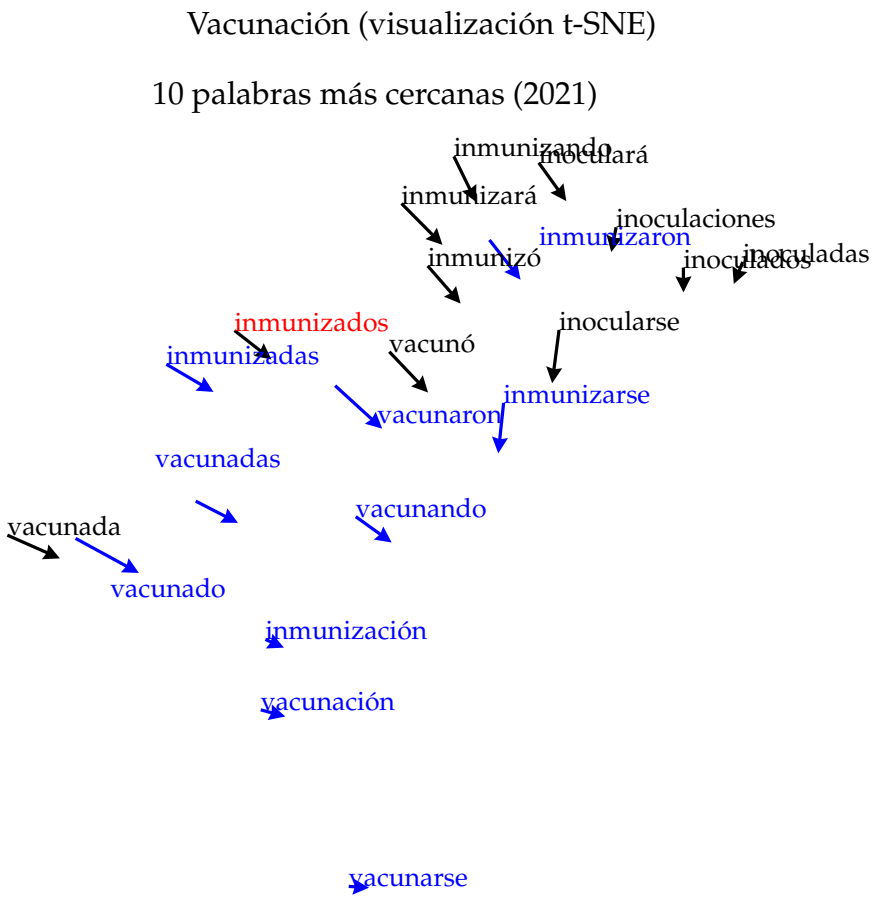

Figura 5.27: Análisis de vecinos más cercanos en el 2018 al clúster sobre vacunacion

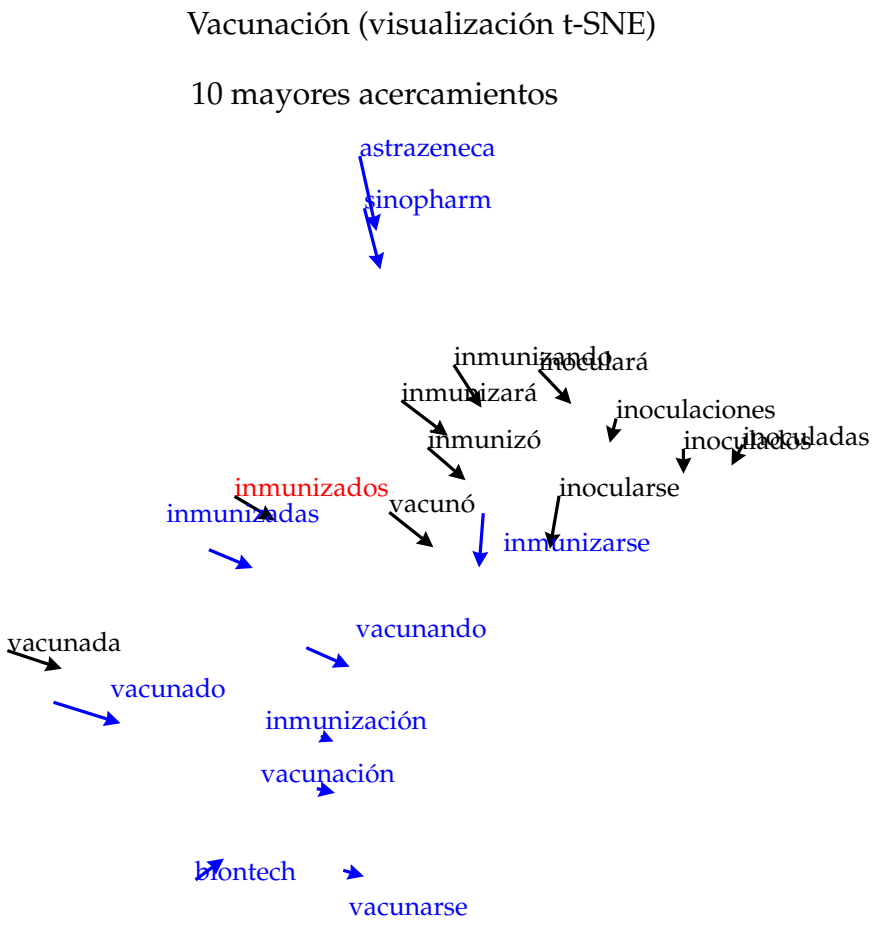

Figura 5.28: Análisis de mayores acercamientos al clúster sobre vacunacion

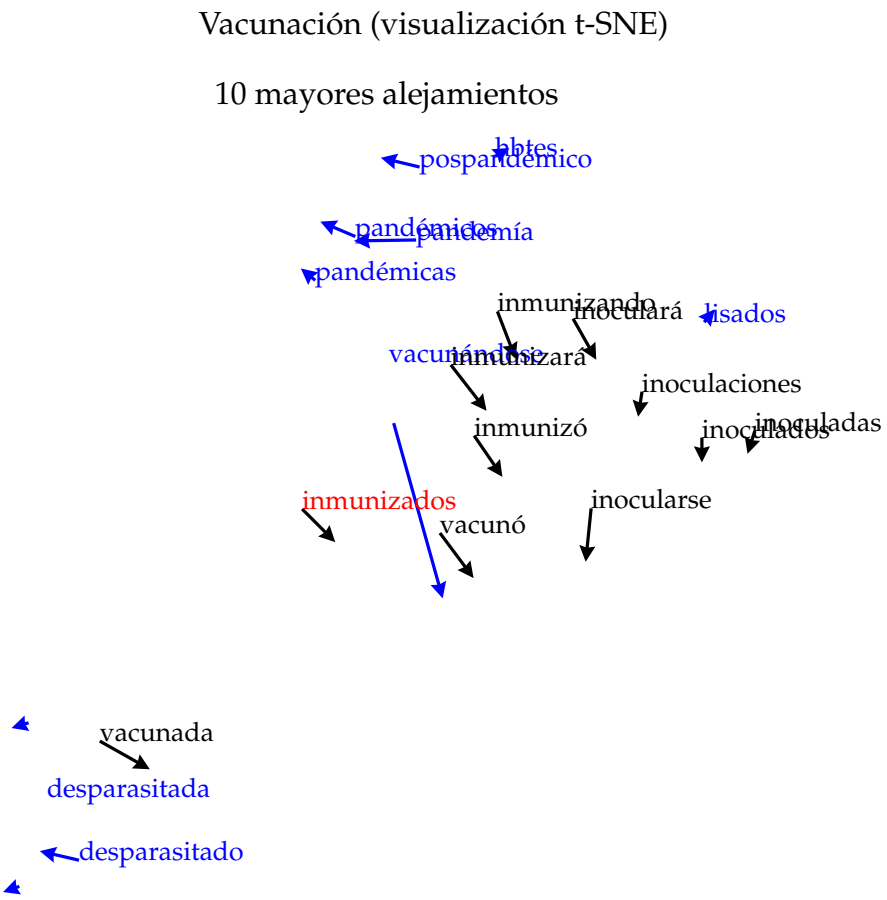

Figura 5.29: Análisis de mayores alejamientos al clúster sobre vacunacion

Para el clúster de vacunación podemos observar que el cambio fue menor respecto a los otros clústeres estudiados. Sin embargo, podemos ver que hubo desplazamientos importantes en otras palabras relacionadas con vacunas contra el COVID, como nombres de fabricantes (Biontech, Astra Zeneca y Sinopharm).

5.2.5. Síntesis del desplazamiento semántico

A partir de los tres casos de estudio podemos concluir que sí es posible encontrar desplazamientos semánticos significativos en periodos cortos de tiempo. Algunos ejemplos son «coronavirus» que se alejó de otros tipos de virus y se acercó a términos relacionados con COVID-19, o bien las mascarillas que se alejaron de los tratamientos estéticos.

Este tipo de desplazamientos podrían tener implicaciones en las aplicaciones prácticas de los *word embeddings*. Por ejemplo, un motor de búsqueda que use *embeddings* viejos no podría asociar «coronavirus» con «covid», por ejemplo. Presumiblemente, este comportamiento también sucedería en otras áreas como nombres de productos nuevos o personas (deportistas, cantantes, políticos, etc). Esto ejemplifica la necesidad de reentrenar regularmente los *word embeddings* para poder «adquirir nuevo conocimiento» y que sigan siendo útiles.

5.3. Resultados de Desplazamiento Semántico Relativo a Emociones

En la sección anterior se analizaron algunos de los clústeres que tuvieron más desplazamientos semánticos entre el 2018 y el 2021. También es posible investigar qué tanto se desplazaron las palabras respecto a grupos de emociones.

Al calcular el peso emocional absoluto, las diferencias por cada emoción son muy sutiles. Sin embargo, sí se pueden detectar algunos desplazamientos según la temática de las palabras, aunque dichos movimientos son muy pequeños. En este caso se mostrarán los desplazamientos emocionales respecto a los siguientes temas:

1. COVID: El clúster contiene sinónimos de COVID como coronavirus y términos asociados como pandemia o confinamientos. No incluye vacunas.

2. Mascarillas: Incluye sinónimos de mascarillas como cubrebocas, barbijo o tapaboca.

3. Vacunación: Incluye sinónimos de vacunarse, como inoculará o inmunizar.

5.3.1. COVID-19

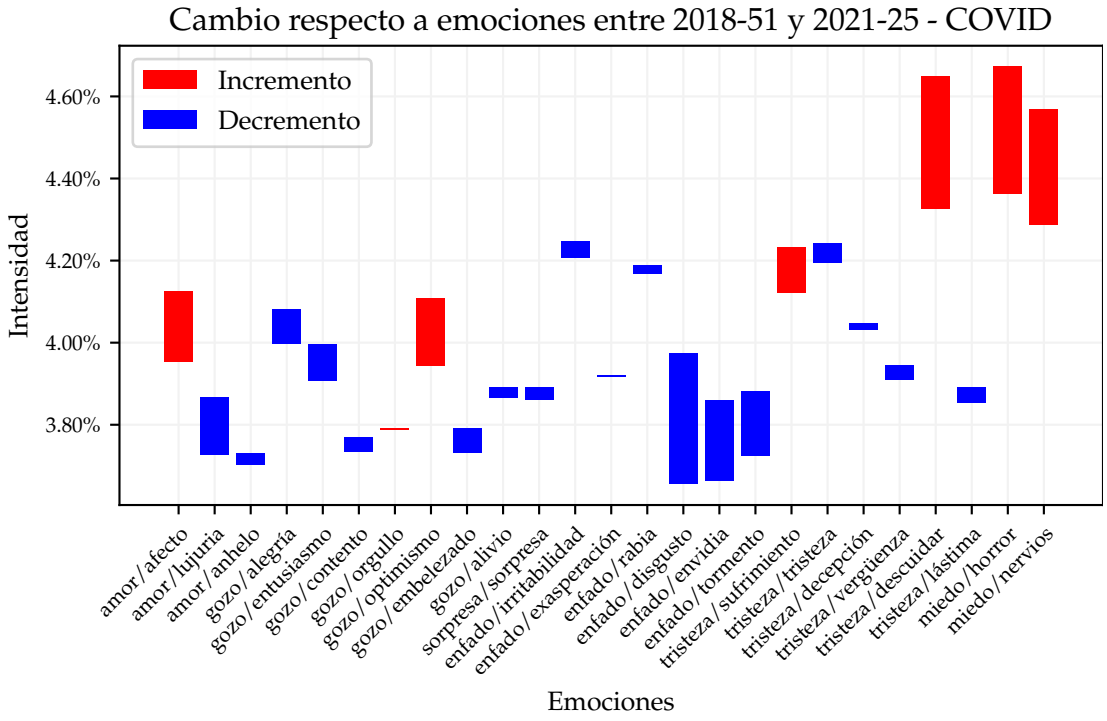

Figura 5.30: Cambio en cercanía a emociones secundarias del clúster «COVID-19»

En el caso del clúster de COVID-19, las emociones dominantes son *fear/horror* (horror), *fear/nervousness* (nerviosismo), *sadness/neglect* (negligencia). Estas están seguidas por *sadness/suffering* (sufrimiento), *sadness/sadness* (tristeza), *anger/irritability* (irritabilidad) y *anger/rage* (furia), tal como se muestra en la Figura 5.30. De la Figura 5.31 en la página siguiente, podemos decir que en general las emociones primarias dominantes del COVID-19 están relacionadas con miedo y luego con tristeza. Además, estas se alejan de las emociones primarias como alegría, sorpresa y enojo.

Como se muestra en la Figura 5.30, los mayores desplazamientos emocionales ocurrieron en *fear/horror*, *fear/nervousness* y *anger/disgust* (disgusto, en este caso se alejó).

Los valores de intensidad emocional por palabra están detallados en la Apéndice C.1 en la página 88.

Al analizar las emociones terciarias relacionadas con miedo, podemos ver en la Figura 5.32 en la página siguiente, que la relación con «miedo» se dio en las emo-

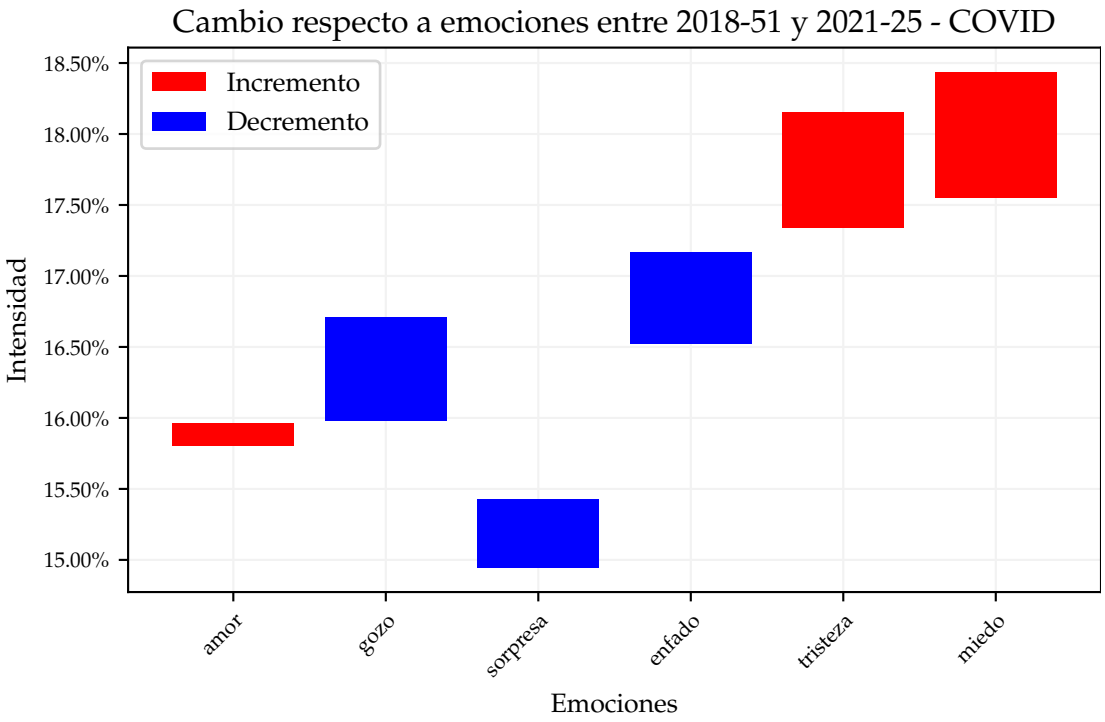

Figura 5.31: Cambio en cercanía a emociones primarias del clúster «COVID-19»

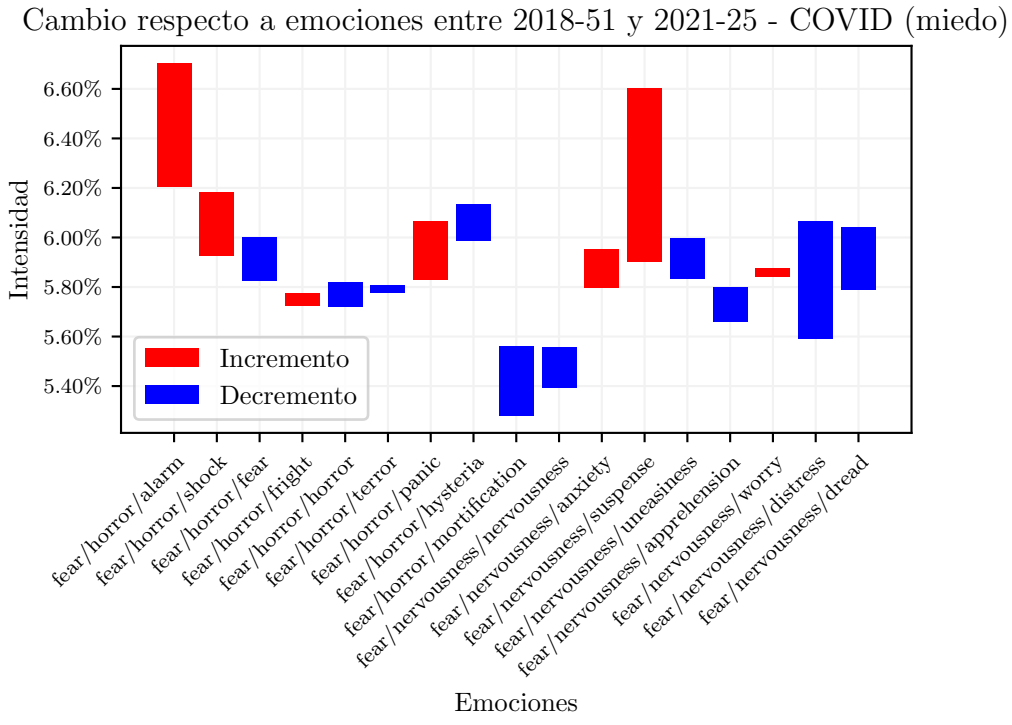

Figura 5.32: Cambio en cercanía a emociones terciarias de miedo del clúster «COVID-19»

ciones terciarias relacionadas con *fear/horror/alarm* (alarma), *fear/nervousness/suspense* (suspenso), *fear/horror/shock*, *fear/horror/panic* (pánico) y *fear/horror/hysteria* (histeria). Además, esta asociación no existía previa a la pandemia: en la misma figura podemos ver cambios significativos en *fear/horror/alarm* y *fear/nervousness/suspense*.

5.3.2. Mascarillas

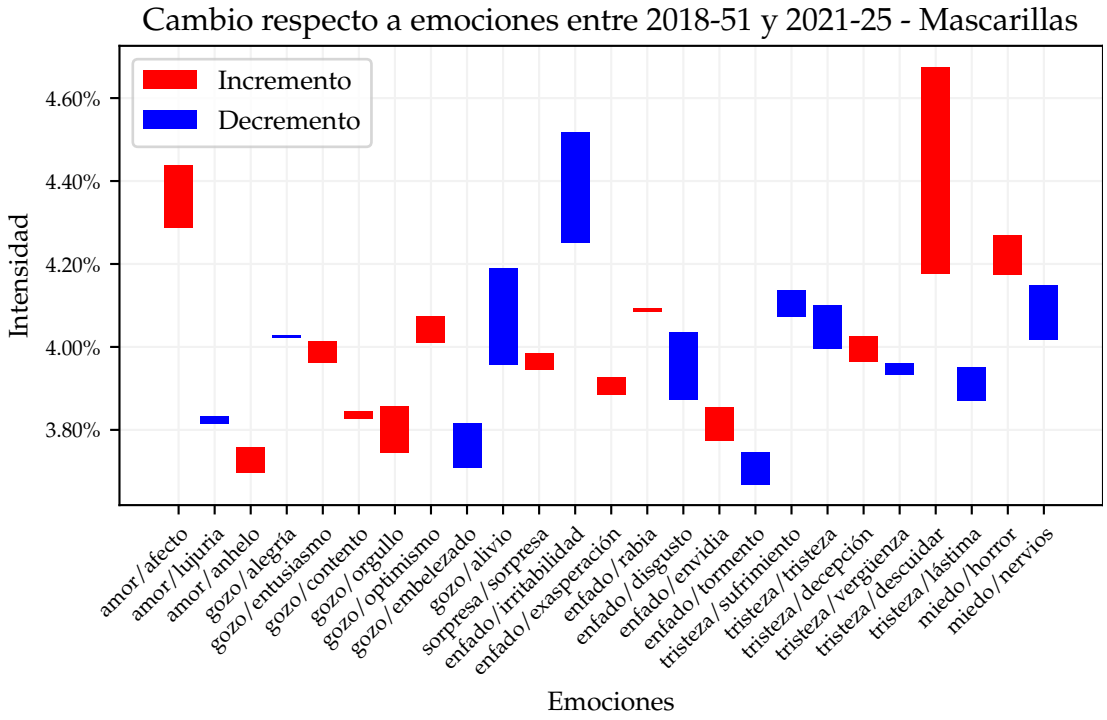

Figura 5.33: Cambio en cercanía a emociones secundarias del clúster «mascarillas»

En el clúster de términos relacionados con mascarillas, podemos ver que las emociones secundarias dominantes son *love/affection*, *anger/irritability*, *sadness/neglect* y *fear/horror*, lo que se muestra en la Figura 5.33. Sin embargo, al medir la diferencia, podemos ver que estas palabras se alejaron de *anger/irritability*, mientras que se acercaron a *love/affection* y a *sadness/neglect*.

Dentro de las emociones terciarias, tenemos que la emoción predominante en *love/affection* es *love/affection/caring*, como se muestra en la Figura 5.34 en la página siguiente. Esta fue la subemoción terciaria con mayor crecimiento, seguida por *love/affection/fondness* y *love/affection/adoration*, como se observa en la Figura 5.34 en la página siguiente. De hecho, acá el mayor acercamiento se da entre la palabra «distanciamiento» y la emoción terciaria *love/affection/caring*.

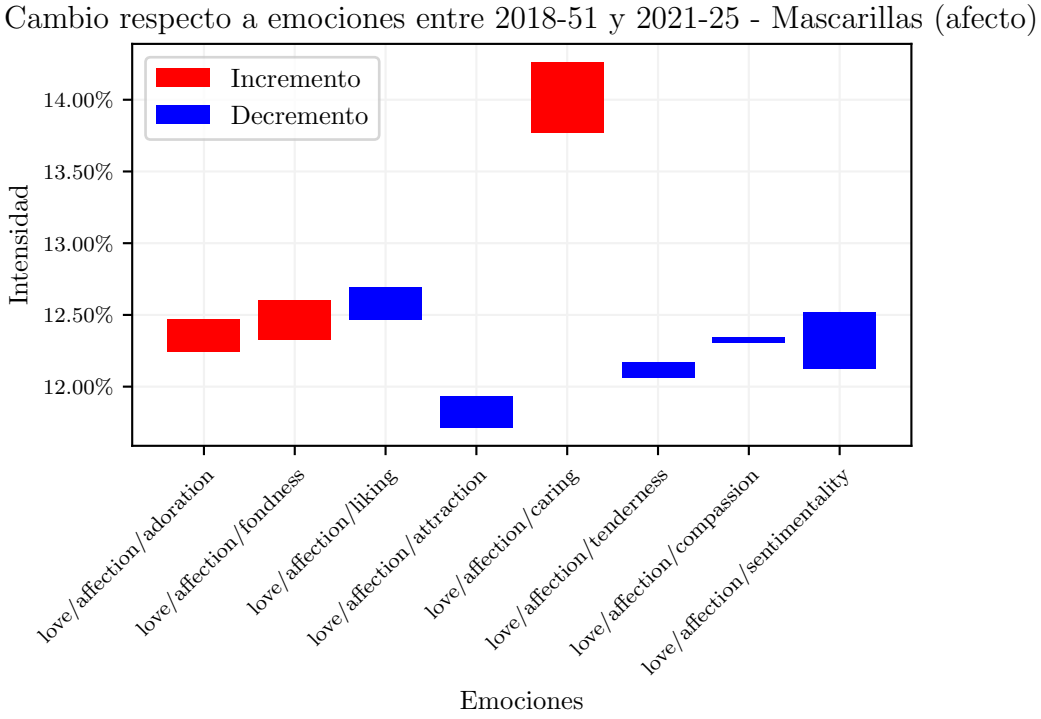

Figura 5.34: Cambio en cercanía a emociones terciarias de afecto del clúster «mascarillas»

Los valores de intensidad emocional por palabra están detallados en la Apéndice C.2 en la página 92.

5.3.3. Vacunación

Respecto al clúster de vacunación, tal como se muestra en la Figura 5.35 en la página siguiente, la emoción secundaria dominante es *fear/horror*, seguida por *anger/rage* y *sadness/neglect*. No obstante, estas no fueron las emociones que más cambiaron. Al parecer el miedo a las vacunas ya era preexistente, ya que el desplazamiento de *fear/horror* fue mínimo (es decir, en el 2018 también era la emoción secundaria dominante). En la misma figura, se muestran los desplazamientos emocionales, donde más bien los mayores acercamientos estuvieron en *love/longing* (anhelar) y *joy/optimism*. Por otro lado, los mayores alejamientos ocurrieron en *anger/irritability* y *anger/disgust*.

Los valores de intensidad emocional por palabra están detallados en la Apéndice C.3 en la página 94.

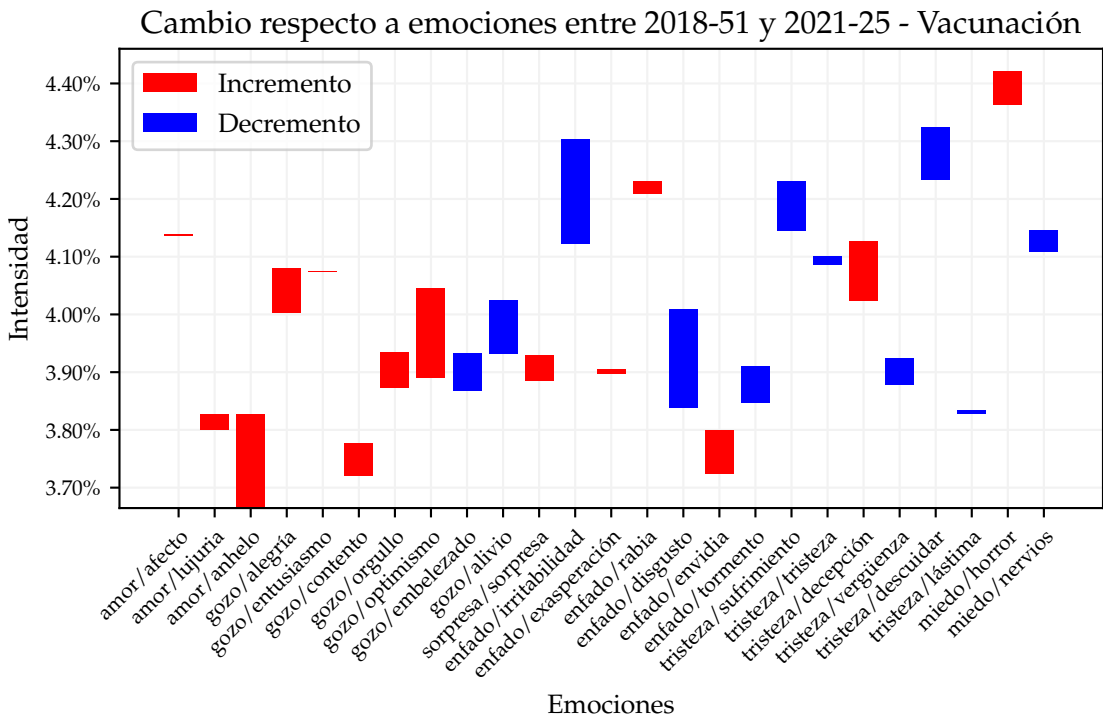

Figura 5.35: Cambio en cercanía a emociones secundarias del clúster «vacunación»

5.3.4. Síntesis del análisis de desplazamiento emocional

El análisis de desplazamiento semántico se hizo basado en el modelo de clasificación de emociones para oraciones propuesto por Alshahrani et al. (2017). Las principales diferencias fueron usar texto y *word embeddings* en español (en lugar de inglés) y usar la clasificación de emociones de Shaver et al. (1987) en lugar de la clasificación de emociones propuesta por Ekman (1971).

En este análisis los desplazamientos obtenidos son muy sutiles, usualmente de menos de 1 % luego de aplicar *softmax* sobre las similitudes de coseno. Sin embargo, en algunos casos estos cambios podrían cambiar los resultados de un clasificador de emociones. Por ejemplo, en el caso de «mascarillas» la emoción secundaria predominante dejó de ser «enojo/irritabilidad» y pasó a ser «tristeza/negligencia». Esto podría ser suficiente para cambiar el resultado de un clasificador de emociones basado en *Word Mover's Distance*, como el propuesto por Alshahrani et al. (2017).

En este caso solamente se analizó el desplazamiento semántico aplicado a la cercanía emocional. Sin embargo, existen muchísimos otros usos donde podría ser aplicable: detección de noticias falsas, discursos de odio, etc.

Capítulo 6

Conclusiones

A continuación se detallan las conclusiones de las cuatro grandes áreas del presente trabajo: descarga del corpus, generación de *word embeddings*, desplazamiento semántico y desplazamiento emocional.

6.1. Descarga de datos

Al descargar los datos es necesario considerar desafíos técnicos como la codificación del texto, idioma, errores, compresión, etc. Usar los datos de CommonCrawl reduce algunos de estos problemas, pues el conjunto de datos ya incluye la codificación e idioma detectados.

Debido a que CommonCrawl se encuentra en AWS S3, el mejor rendimiento posible se logra haciendo la descarga en la misma región. En este caso hay que considerar los costos de transferencia: ya sea calcular todo en la nube de Amazon o bien extraer los datos de CommonCrawl a otro servicio de cómputo. La segunda opción requiere un gran ancho de banda y amplias capacidades de almacenamiento. La primera requiere planeamiento, ya que habría que pagar recursos de cómputo, almacenamiento y transferencia fuera de AWS. Es de suma importancia analizar los costos antes de empezar los cálculos, ya que estos potencialmente podrían ser muy elevados.

6.2. Generación de *word embeddings*

El algoritmo de *word embeddings* y su configuración afectan el tipo de clústeres que se generan. Por ejemplo, una ventana grande produce *word embeddings* relacio-

nados los temas tratados, mientras que una ventana pequeña captura información de la palabra (Levy y Goldberg, 2014).

Según el volumen de datos por procesar, algunas implementaciones de generación de *word embeddings* pueden no funcionar correctamente. En este trabajo se encontraron implementaciones que requirieron ser modificadas para soportar valores de más de 32 bits (como pWord2Vec) o bien soluciones que no eran capaces de escalar al añadir más CPU (como Wego o gensim).

En el caso de gensim, esta fue la única solución evaluada que soporta texto comprimido con *GZip*. Sin embargo, esto reduce significativamente el rendimiento. La alternativa, usada en este trabajo, fue comprimir el texto usando un diccionario ordenado por frecuencia. Esto redujo el texto significativamente y no requirió desperdiciar CPU descomprimiendo el texto. Por lo tanto, esta es una buena opción para comprimir texto del cual se va a entrenar un modelo de *word embeddings*.

También es necesario considerar el costo o rendimiento de las implementaciones de *word embeddings*. Una implementación como gensim puede ser más sencilla de utilizar, pero puede ser hasta 20 veces más lenta que pWord2Vec en hardware similar. Si se está pagando por recursos en la nube, o bien se está consumiendo una cuota de cómputo limitada de un clúster universitario compartido, este es un aspecto que se debe tomar en cuenta.

6.3. Desplazamiento semántico

El desplazamiento semántico por creación de entidades debe ser tomado en cuenta al crear aplicaciones que usen *word embeddings*, pues es imposible que el modelo conozca la nueva naturaleza de la palabra. En este trabajo se detectaron algunos términos que cambiaron significativamente de vecinos: COVID, coronavirus, vacunas, etc. Estos cambios abruptos sucedieron en un periodo corto de tiempo. Por lo tanto, se confirma que las aplicaciones que usan *word embeddings* requieren que estos sean reentrenados constantemente para que el modelo «aprenda» sobre los cambios en los usos de las palabras.

6.4. Desplazamiento emocional

Respecto al desplazamiento emocional, en este trabajo se encontró que los desplazamientos son sutiles. Sin embargo, esto podría ser suficiente para cambiar las categorías obtenidas de un clasificador de emociones.

6.5. Trabajo futuro

A continuación se indican diversas oportunidades de mejora en el presente trabajo, así como otras áreas de investigación que se podrían realizar a partir de este.

Una limitación importante del trabajo fue usar *word2vec* en lugar de otro tipo de *word embeddings* que sea sensible al contexto como *BERT*. Recientemente, se han publicado tesis donde se investiga el uso de *BERT* para análisis de desplazamiento semántico, como por ejemplo Montariol (2021). Sin embargo, el costo de entrenar *BERT* desde cero es prohibitivamente caro, así que para repetir este estudio usando *word embeddings* sensibles al contexto se requiere un financiamiento mucho mayor, reducir el costo de entrenamiento de *BERT* o bien desarrollar otra técnica de *word embeddings* con un costo de entrenamiento menor.

Otra mejora posible es realizar un proceso de detección de frases, para poder identificar términos compuestos como «Costa Rica». Por otro lado, también se podría hacer uso de un lematizador para reducir las formas flexionadas al lema y así reducir la cantidad de palabras procesadas.

Para hacer este trabajo se procesó una gran cantidad de datos que no es posible examinar en su totalidad de forma detallada. Detrás de cada desplazamiento semántico hay una historia que lo justifica y esto puede ser interesante para diversas áreas del conocimiento. Investigar las razones de detrás de estos cambios pueden ser un área de estudio interesante por sí misma, ya sea por motivos lingüísticos, históricos, de estudio de mercado, etc.

Este trabajo se limitó a estudiar los desplazamientos entre dos puntos en el tiempo: la semana 51 del 2018 y la semana 25 del 2021. Sin embargo, gracias a colecciones como *CommonCrawl* es posible acceder a textos de Internet recolectados aproximadamente cada mes desde el 2011. Las optimizaciones mencionadas en este trabajo pueden servir de guía para facilitar y reducir los costos de elaborar un trabajo similar a mayor escala que estudie más puntos en el tiempo.

El análisis de emociones usado en este trabajo se basó en la clasificación jerárquica de Shaver et al. (1987). Sin embargo, esta clasificación fue elaborada utilizando *agrupamiento* de palabras en inglés y la opinión de personas angloparlantes. Para elaborar este trabajo se realizó una traducción propia de las palabras, pero en algunos casos no se encontraron traducciones apropiadas o bien una misma palabra se podía asociar a dos emociones diferentes. En esta situación, lo ideal sería elaborar una clasificación emocional específica para el español, la cual sí puede estar basada en la misma metodología de Shaver et al. (1987). Esto podría mejorar los análisis

emocionales al contemplar emociones que existan en un idioma, pero no puedan ser traducidas al inglés.

Apéndice A

Algoritmo de compresión

En este capítulo se muestra una implementación en Python del algoritmo de compresión utilizado para almacenar el corpus. Su principal característica es que no es necesario descomprimir el texto para ejecutar algoritmos como *Word2Vec,* TF-IDF o conteo de frecuencias. Al evitar este paso se reduce el tiempo de procesamiento.

Código Fuente A.1: Implementación en Python del algoritmo de compresión usado

```
base62_digits = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
base62_value = {c:i for i,c in enumerate(base62_digits)}
used_words = load_words_sorted_by_descending_frequency()
word_to_idx = {w: i+1 for i, w in enumerate(used_words)}

def to_base62(num):
    out = []
    if num == 0:
        return "0"
    while num > 0:
        out.append(base62_digits[int(num % 62)])
        num //= 62
    out.reverse()
    return ''.join(list(out))

def word2key(word):
    if word in word_to_idx:
        return to_base62(word_to_idx[word])
    else:
```

```
20          return f"!{word}"
21
22  def key_to_word(key):
23      if key[0] == "!":
24          return key[1:]
25      value = 0
26      for c in key:
27          value *= 62
28          value += base62_value[c]
29      return used_words[value-1]
30
31  def compress(data):
32      lines = [line for line in data.split("\n")]
33      lines = [" ".join([word2key(w) for w in line.split()]) for line in lines]
34      return "\n".join(lines)
35
36  def decompress(data):
37      lines = [line for line in data.split("\n")]
38      lines = [" ".join([key_to_word(w) for w in line.split()]) for line in lines]
39      return "\n".join(lines)
```

Apéndice B

Listado de Palabras por Emoción

Para medir la similitud de cada palabra, cada emoción se utilizó la clasificación de emociones propuesta por Shaver et al. (1987). Como esta clasificación jerárquica está en inglés, fue necesario realizar una traducción. A continuación, se listan las emociones de la jerarquía de Shaver et al. (1987) sin traducir y finalmente la traducción usada en este trabajo. Algunas de las traducciones tienen poco uso en español o bien son absurdas (como «anhelarse» o «disfrutación»), sin embargo, se conservaron para mantener en la medida de lo posible los mismos tipos de variaciones en las listas. En algunas emociones se indica «N/A» porque no se encontraron traducciones apropiadas o bien ya habían sido incluidas en otras sub emociones.

1. *love*
    - *a*) *affection*
        1) *adoration*: adoración, adorar, adorable, adorado, adorada, adorando, adorarse
        2) *fondness*: cariño, cariñoso, encariñar, encariñable, encariñado, encariñada, encariñando, encariñarse
        3) *liking*: gusto, gustar, gustable, gustado, gustada, gustando, gustarse
        4) *attraction*: atracción, atraer, atraíble, atraído, atraída, atrayendo, atraerse
        5) *caring*: cuido, cuidar, cuidable, cuidado, cuidada, cuidando, cuidarse
        6) *tenderness*: tierno, ternura
        7) *compassion*: compasión, compasivo, compasiva, compasible, compadecer, compadecerse, compadeciendo

8) *sentimentality*: sentimentalismo, sentimental, sentimentalista

b) *lust*

1) *lust*: lujuria, lujuriar, lujuriado, lujuriada, lujuriando, lujuriarse
2) *desire*: deseo, desear, deseable, deseado, deseada, deseando, desearse
3) *passion*: pasión, apasionar, apasionable, apasionado, apasionada, apasionando, pasionando, pasionable, pasionado, pasionada, apasionarse
4) *infatuation*: encaprichar, encapricache, encaprichado, encaprichada, encaprichando, encapricharse

c) *longing*

1) *longing*: anhelo, anhelar, anhelable, anhelado, anhelada, anhelando, anhelarse

2. *joy*

a) *cheerfulness*

1) *cheerfulness*: alegre, alegría, alegrable, alegrado, alegrada, alegrando, alegrar, alegrarse
2) *amusement*: diversión, divertir, divertible, divertido, divertida, divirtiendo, divertirse
3) *bliss*: dicha, dichoso, dichosa
4) *gaiety*: N/A
5) *glee*: N/A
6) *jolliness*: N/A
7) *joviality*: jovial, jovialidad, joviable
8) *joy*: gozo, gozar, gozando, gozado, gozada, gozarse
9) *delight*: deleitación, deleite, deleitar, deleitable, deleitado, deleitada, deleitando, deleitarse
10) *enjoyment*: disfrutación, disfrite, disfrutar, disfrutable, disfrutado, disfrutada, disfrutando, disfrutarse
11) *gladness*: encantación, encanto, encantar, encantable, encantado, encantada, encantando, encantarse
12) *happiness*: felicidad, feliz
13) *jubilation*: júbilo, jubiloso, jubilosa

14) *elation*: alborozo, alborozar, alborozado, alborozada, alborozante, alboración, alborozando, alborozarse
15) *satisfaction*: satisfacción, satisfecho, satisfecha, satisfizo, satisfacer, satisfaciendo, satisfacerse
16) *ecstasy*: éxtasis, éxtasi
17) *euphoria*: euforia, eufórico, eufórica

b) *zest*

1) *enthusiasm*: entusiasmo, entusiasmar, entusiasmable, entusiasmado, entusiasmada, entusiasmando, entusiasmarse
2) *zeal*: N/A
3) *excitement*: exitación, exitar, exitable, exitante, exitando, exitada, exitado, exitarse
4) *thrill*: emoción, emocionar, emocionante, emocionado, emocionada, emocionando, emocionarse
5) *exhilaration*: regocijo, regocijar, regocitante, regocijado, regocijada, regocijando, regocijarse

c) *contentment*

1) *contentment*: contento, contentamiento, contentar, contentado, contentada, contentando, contentarse
2) *pleasure*: plácido, placentero, placer, plácida, placido, placiendo, placerse

d) *pride*

1) *pride*: orgullo, orgulloso, orgullecer, enorgullecer, orgulleciendo, orgullecida, orgullecido, orgullecerse
2) *triumph*: triunfo, triunfante, triunfar, triunfado, triunfada, triunfando, triunfarse

e) *optimism*

1) *optimism*: optimismo, optimista, optimizable, optimizando, optimizado, optimizada
2) *eagerness*: afán, afanoso, afanar, afanado, afanada, afanando, afanarse
3) *hope*: esperanza, esperanzar, esperanzante, esperanzado, esperanzada, esperanzando, esperanzarse

f) *enthrallment*

1) *enthrallment*: embelesamiento, embelezado, embelezar, embelezada, embelezando, fascinación, fascinar, fascinable, fascinado, fascinada, fascinando, fascinarse
2) *rapture*: arrebato, arrebatar, arrabatado, arrebatada, arrebatación, arrebatarse

g) *relief*

1) *relief*: aliviación, alívio, aliviar, aliviable, aliviado, aliviada, aliviando, aliviarse

3. *surprise*

a) *surprise*

1) *surprise*: sorpresa, sorprender, sorprendido, sorprendida, sorprendiendo, sorprendente, sorpresivo, sorprenderse
2) *amazement*: asombro, asombrar, asombrado, asombrada, asombrando, asombroso, asombrante, asombrarse
3) *astonishment*: increíble

4. *anger*

a) *irritability*

1) *irritability*: irritación, irritabilidad, irritar, irritable, irritado, irritada, irritando, irritarse
2) *aggravation*: agravación, agravar, agravado, agravada, agravando, agravante, agravarse
3) *agitation*: agitación, agitar, agitado, agitada, agitando, agitante, agitarse
4) *annoyance*: molestia, molestar, molestado, molestada, molesto, molesta, molestando, molestante, molestarse
5) *grouchy*: malhumorado, malhumorada, malhumorar, malhumorante
6) *grumpy*: gruñón, gruñona, gruñir, gruñado, gruñada, gruñante
7) *crosspatch*: cascarrabias

b) *exasperation*

1) *exasperation*: exasperación, exasperar, exasperable, exasperado, exasperada, exasperando, exasperarse

2) *frustration*: frustración, frustrar, frustrante, frustrable, frustrado, frustrada, frustrando, frustrarse

c) *rage*

1) *anger*: enfado, enfadar, enfadado, enfadada, enfadando, enfadable, enfadante, enfadarse
2) *outrage*: indignación, indignar, indignado, indignada, indignante, indignando, indignable, indignarse
3) *fury*: furia, furioso, furiosa, fúrico, fúrica, enfuriarse
4) *wrath*: ira, iracundo, iracunda
5) *hostility*: hostil, hostilidad, hostilmente, hostiles
6) *ferocity*: feroz, ferozmente, ferocidad, feroces
7) *bitterness*: amargura, amargar, amargado, amargada, amargante, amargando, amargable, amargarse
8) *hatred*: odio, odiar, odiado, odiada, odiando, odiante, odiable, odiarse
9) *scorn*: desdén, desdeñar, desdeño, desdeñada, desdeñado, desdeñante, desdeñando, desdeñable, desdeñarse
10) *spite*: despecho, despechar, despechado, despechada, despechante, despechando, despechable, despecharse
11) *vengefulness*: venganza, vengar, vengado, vengada, vengante, vengativo, vengador, vengación, vengable, vengarse
12) *dislike*: disgusto, disgustar, disgustable, disgustada, disgustado, disgustante, disgustarse
13) *resentment*: resentimiento, resentir, resentido, resentida, resentimiento, resentible, resintiéndose, resintiendo, resentirse

d) *disgust*

1) *revulsion*: repugnancia, repugnante, repugnar, repugnable, repugnarse, repugnado, repugnada, repugnando
2) *contempt*: desprecio, despreciar, despreciado, despreciada, despreciarse, despreciando, despreciación, despreciable
3) *loathing*: aborrecer, aborrecido, aborrecida, aborrecible, aborrecerse, aborreciendo

e) *envy*

1) *envy*: envidia, envidiar, envidiado, envidiada, envidiarse, envidiable

2) *jealousy*: celos, celar, celado, celada, celarse, celable

f) *torment*

1) *torment*: tormento, atormentar, atormentarse, atormentado, atormentada, atormentando

5. *sadness*

a) *suffering*

1) *suffering*: sufrir, sufrido, sufrida, sufrimiento, sufrible, sufrirse, sufriente
2) *agony*: agonía, agonizar, agonizante, agonizado, agonizada, agonizando
3) *anguish*: angustia, angustiar, angustiante, angustiarse, angustiado, angustiada, angustiando
4) *hurt*: herir, herido, herida, herirse, hiriendo, hiriente

b) *sadness*

1) *sadness*: tristeza, triste
2) *depression*: depresión, deprimir, deprimido, deprimida, deprimiendo, deprimente, deprimirse
3) *despair*: desesperación, desesperar, desesperado, desesperada, desesperarse, desesperante
4) *gloom*: pesimismo, pesimista
5) *glumness*: N/A
6) *unhappiness*: infelicidad, infeliz, descontento, descontenta, descontentarse, descontentar, descontentando, descontentante
7) *grief* : duelo, aflicción, quebranto, quebrantarse, quebrantado, quebrantada, afligirse, afligido, afligida
8) *sorrow*: pena, tormento
9) *woe*: congoja, acongojado, acongojada, acongojante, acongojación, acongojar, acongojarse
10) *misery*: miseria, miserable, desdicha, desdichado, desdichada
11) *melancholy*: melancolía, melancolizar, melancólico, melancólica, melancolizando, melancolizarse, melancolizante

c) *disappointment*

1) *disappointment*: decepción, decepcionar, decepcionado, decepcionada, decepcionante, decepcionarse
2) *dismay*: consternación, consternar, consternado, consternada, consternante, consternarse, desaliento, desalienta, desalentar, desalentado, desalentada, desalentado, desalentarse, desalentando
3) *displeasure*: desagradada, desagradado, desagradar, desagardarse, desagradecido, desagradecida

d) *shame*

1) *shame*: vergüenza, vergonzoso, vergonzosa, avergonsante, avergonzar, avergonzando, avergonzarse
2) *guilt*: culpa, culpar, culposo, culpado, culpada, culparse, culpando
3) *regret*: arrepentirse, arrepentimiento, arrepentir, arrepentido, arrepentida, arrepintiendo
4) *remorse*: remordimiento, remorder, remordido, remordida, remorderse, remordimiendo

e) *neglect*

1) *neglect*: descuidar, descuidado, descuidada, descuidarse, descuidando
2) *alienation*: alienación, alienado, alienada, alienando, alienándose, alejar, alejado, alejada, alejando, alejándose
3) *defeatism*: derrotismo, derrotista
4) *dejection*: abatimiento, abatir, abatido, abatida, abatirse, abatiendo
5) *embarrassment*: N/A
6) *homesickness*: nostalgia, nostálgico, nostálgica
7) *humiliation*: humillación, humillar, humillado, humillada, humillante, humillarse, humillándose
8) *insecurity*: inseguridad, inseguro
9) *insult*: insulto, insultado, insultada, insultar, insultarse, insultándose
10) *isolation*: aislado, aislar, aislada, aislándose, aislando, aislamiento
11) *loneliness*: soledad
12) *rejection*: rechazo, rechazar, rechazado, rechazada, rechazando, rechazarse

f) *sympathy*

1) *pity*: lástima, lastimera, lastimable

2) *mono no aware*: efímero, efímera, impermanencia, brevedad, breve
3) *sympathy*: simpatía, simpático, simpática, simpatizar, simpatizante, simpatizarse, simpatizado, simpatizada, simpatizando

6. *fear*

a) *horror*

1) *alarm*: alarma, alarmar, alarmado, alarmada, alarmarse, alarmando, alarmable
2) *shock*: *shock*, conmoción, conmocionar, conmocionado, conmocionada, conmocionarse, conmocionante, conmocionando, conmocionable
3) *fear*: miedo, temer, temeroso, temido, temida, temerse, temiendo, temible
4) *fright*: susto, asustar, asustado, asustada, asustarse, asustando, asustable
5) *horror*: horror, horrorozo, horroroza, horrible, horripilante, horrorizar, horrorizado, horrorizada, horrorizarse, horrorizante, horrorizando, horrorizable
6) *terror*: terror, terrorífico, terrorífica, aterrorizar, aterrorizado, aterrorizada, aterrizarse, aterrorizable, aterrorizando, aterrorizante
7) *panic*: pánico, paniquear, paniqueado, paniqueada, paniquearse
8) *hysteria*: histeria, histérico, histérica
9) *mortification*: mortificación, mortificar, mortificado, mortificada, mortificarse, mortificando, mortificable

b) *nervousness*

1) *nervousness*: nervioso, nervios, nerviosa
2) *anxiety*: ansiedad, ansioso, ansiosa, ansias
3) *suspense*: suspenso, incertidumbre
4) *uneasiness*: inquietud, desasosiego, intranquilidad, desasosiega, intranquilo, intranquila, inquieto, inquieta
5) *apprehension*: aprensión
6) *worry*: preocupación, preocupar, preocupado, preocupada, preocuparse, preocupando, preocupándose, preocupable
7) *distress*: peligro, peligrar, peligroso, peligrosa, peligrar, peligrando, peligrado, peligrada

8) *dread*: pavor, pavoroso, pavorosa

Apéndice C

Cercanía a emociones en palabras de los clústeres

En la Apartado 5.3 en la página 67, se mostraron resultados agregados sobre los desplazamientos de los clústeres estudiados. Sin embargo, dichos datos fueron generados primero para cada palabra del clúster. Este capítulo incluye dichos datos sin agregar.

Para cada tema se incluyen tres gráficos: el primero con la intensidad o cercanía a emociones secundarias en el 2018 y en el 2021. Luego otro gráfico con la diferencia entre ambos años, donde un valor positivo representa que hubo un acercamiento y uno negativo que hubo un alejamiento.

C.1. COVID

Para el clúster «COVID-19», se incluyeron tres figuras con los siguientes datos detallados por palabra. En la Figura C.1 en la página siguiente se muestra la intensidad emocional de cada palabra respecto a cada emoción secundaria para el año 2018. En la Figura C.2 en la página 90 se muestra la intensidad emocional de cada palabra respecto a cada emoción secundaria para el año 2021. Finalmente, en la Figura C.3 en la página 91 se muestra la diferencia entre los valores del 2021 con los del 2018, de forma que un valor positivo implica que hubo un acercamiento y un valor negativo implica que hubo un alejamiento.

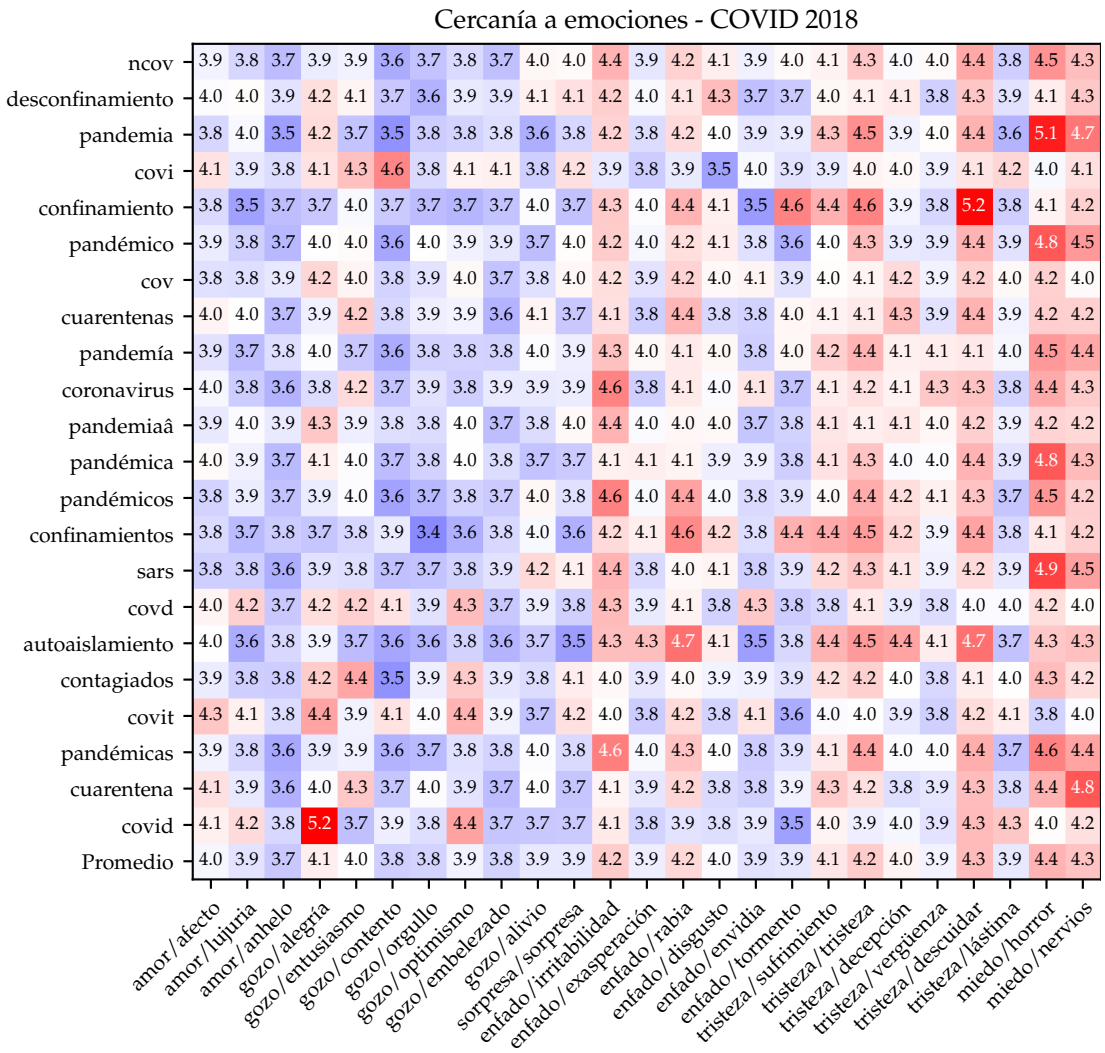

Figura C.1: Cercanía a emociones del clúster «COVID-19 (2018)»

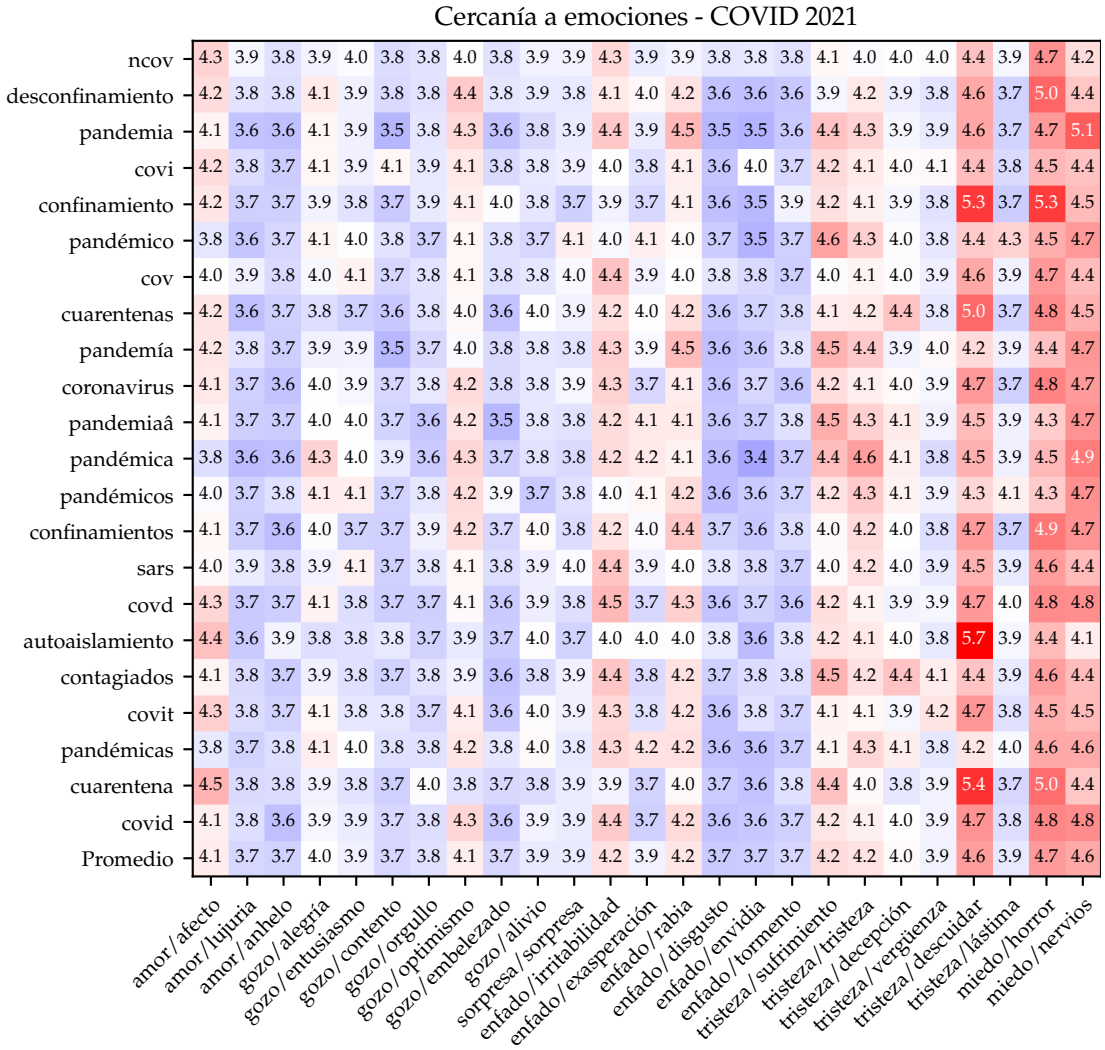

Figura C.2: Cercanía a emociones del clúster «COVID-19 (2021)»

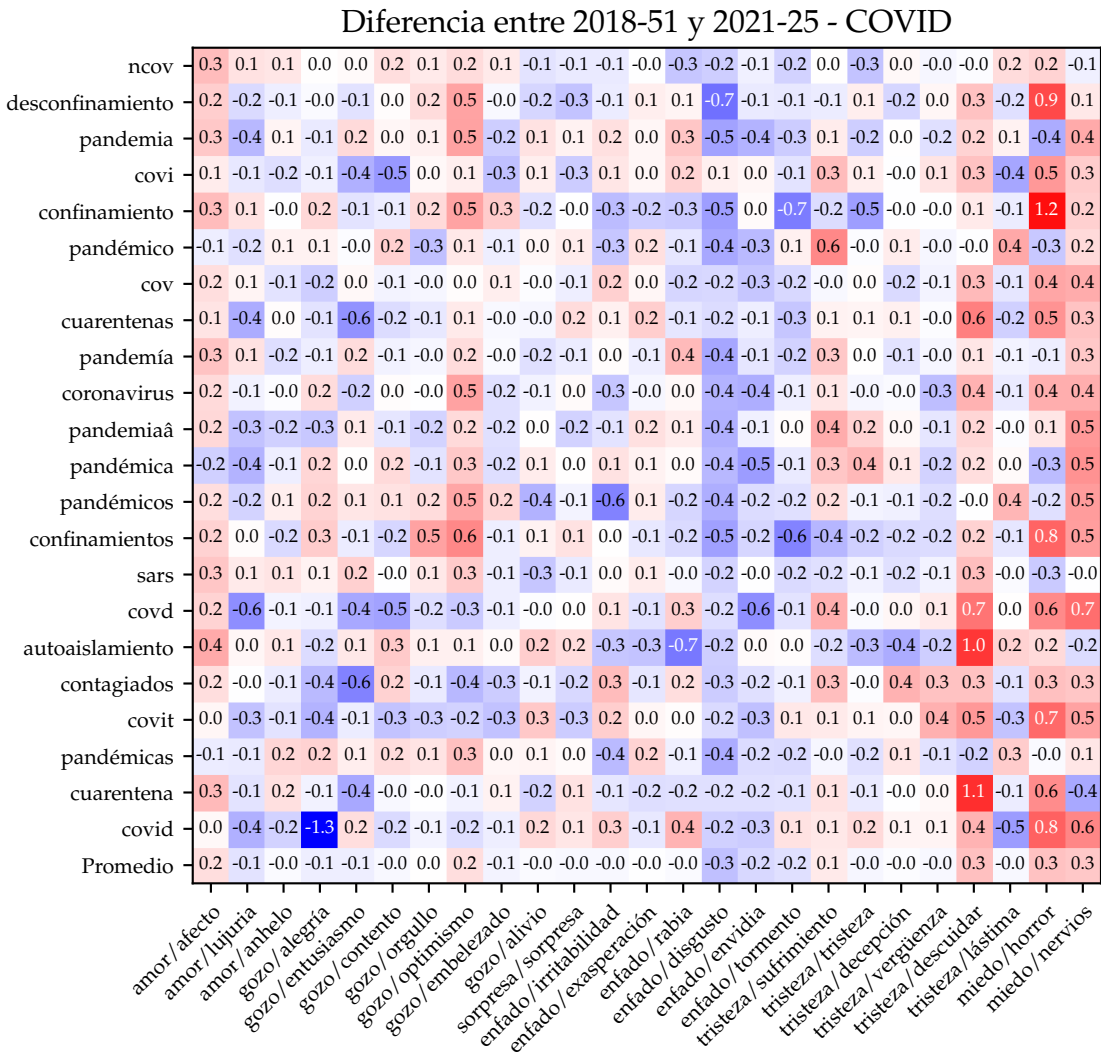

Figura C.3: Cambio en cercanía a emociones del clúster «COVID-19»

C.2. Mascarillas

Para el clúster «mascarillas», se incluyeron tres figuras con los siguientes datos detallados por palabra. En la Figura C.4 se muestra la intensidad emocional de cada palabra respecto a cada emoción secundaria para el año 2018. En la Figura C.5 en la página siguiente se muestra la intensidad emocional de cada palabra respecto a cada emoción secundaria para el año 2021. Finalmente, en la Figura C.6 en la página siguiente se muestra la diferencia entre los valores del 2021 con los del 2018, de forma que un valor positivo implica que hubo un acercamiento y un valor negativo implica que hubo un alejamiento.

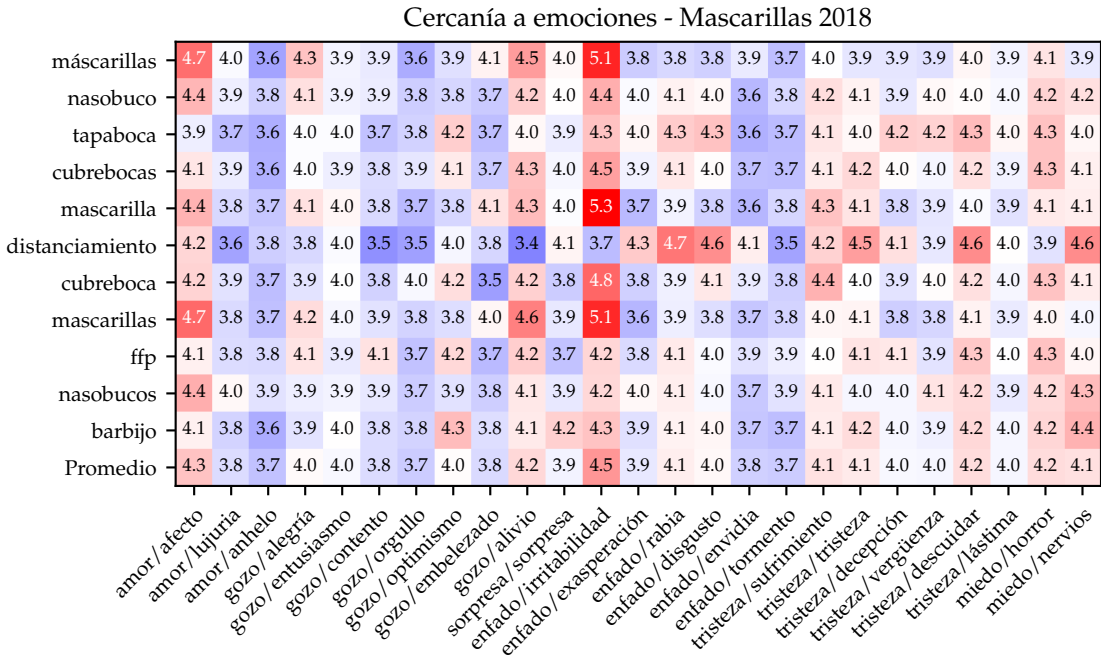

Figura C.4: Cercanía a emociones del clúster «mascarillas (2018)»

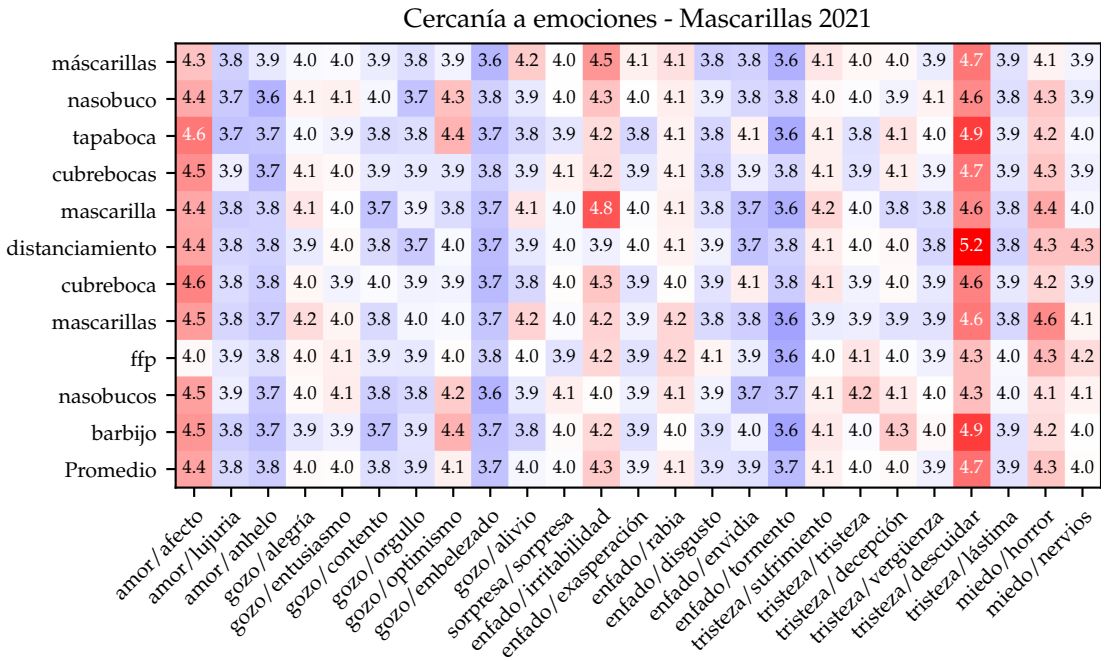

Figura C.5: Cercanía a emociones del clúster «mascarillas (2021)»

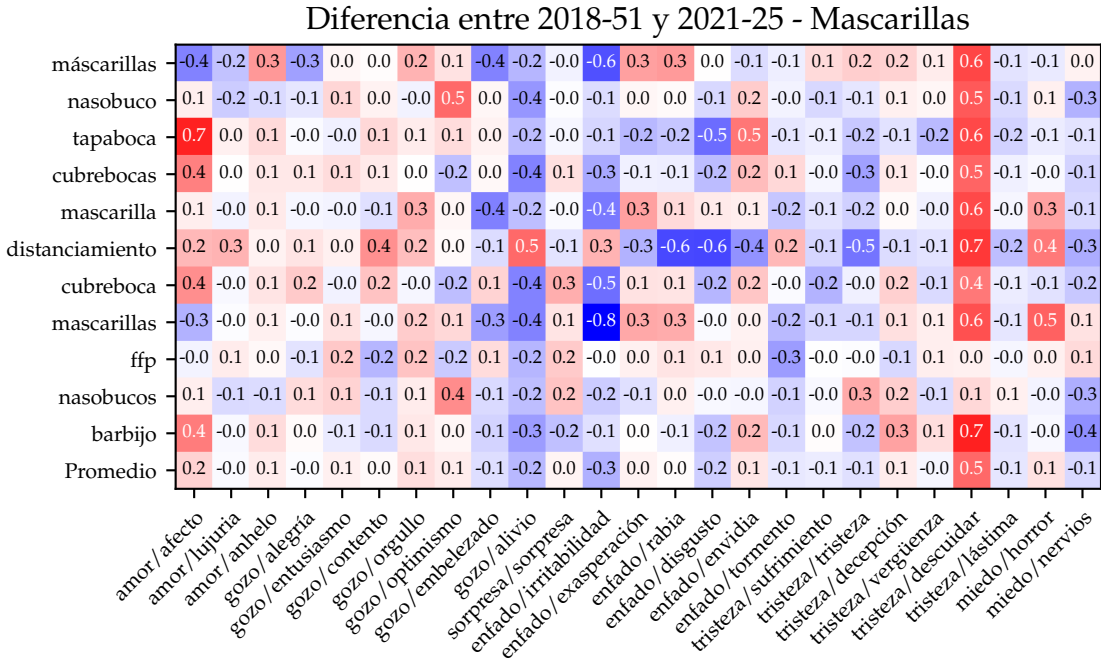

Figura C.6: Cambio en cercanía a emociones del clúster «mascarillas»

C.3. Vacunas

Para el clúster «vacunación», se incluyeron tres figuras con los siguientes datos detallados por palabra. En la Figura C.7 se muestra la intensidad emocional de cada palabra respecto a cada emoción secundaria para el año 2018. En la Figura C.8 en la página siguiente se muestra la intensidad emocional de cada palabra respecto a cada emoción secundaria para el año 2021. Finalmente, en la Figura C.9 en la página 96 se muestra la diferencia entre los valores del 2021 con los del 2018, de forma que un valor positivo implica que hubo un acercamiento y un valor negativo implica que hubo un alejamiento.

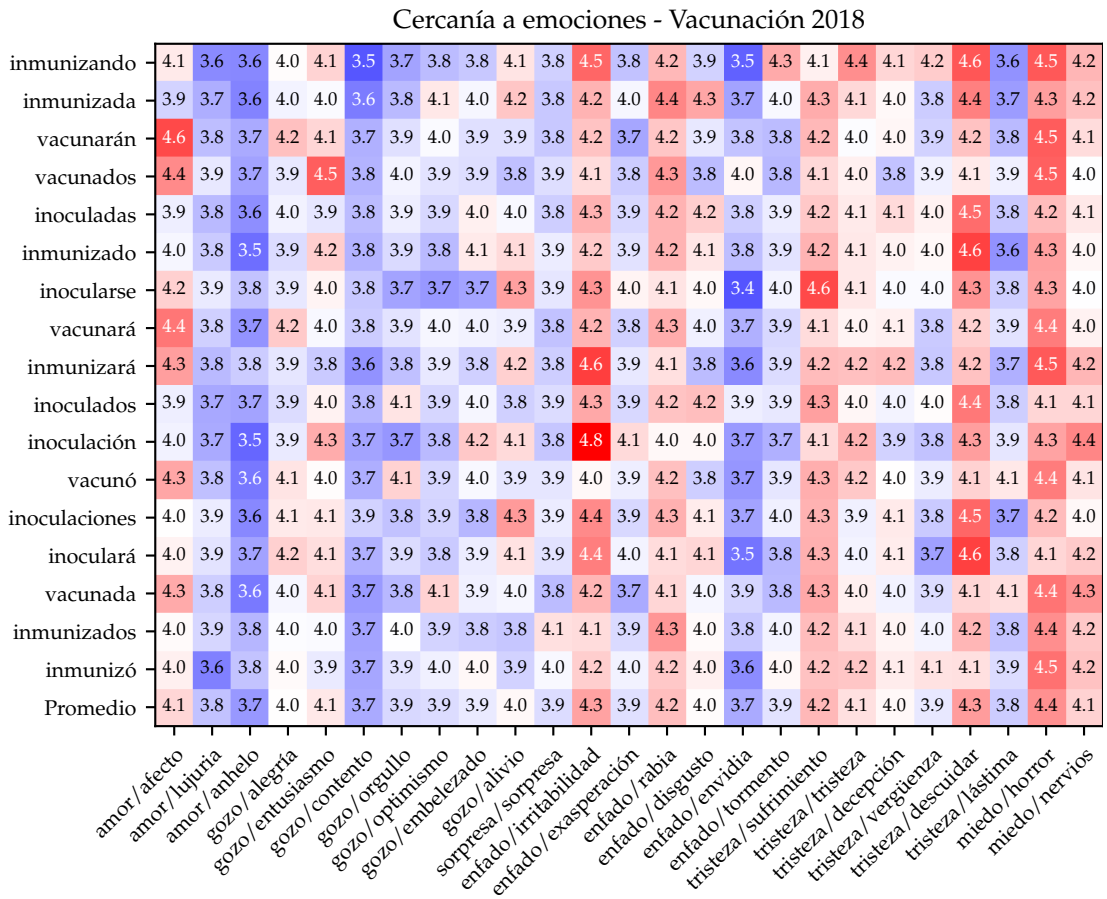

Figura C.7: Cercanía a emociones del clúster «vacunación (2018)»

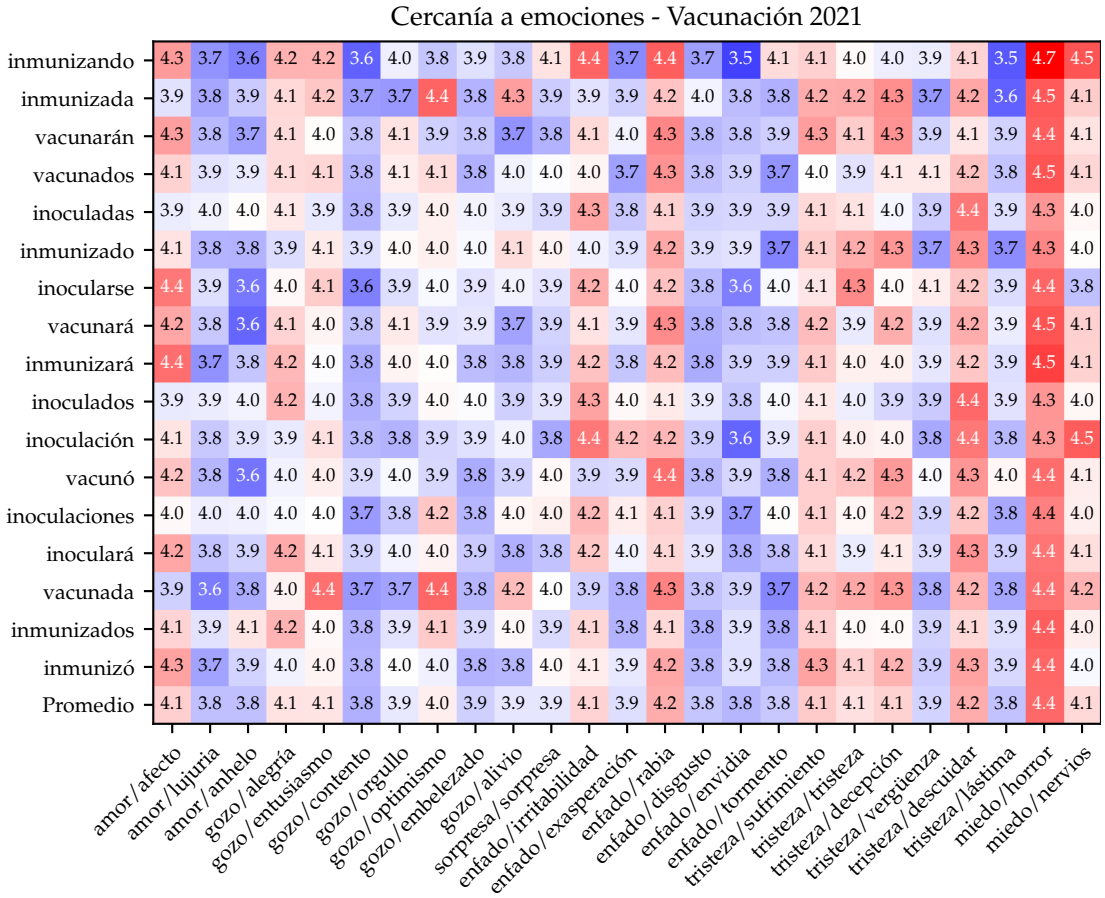

Figura C.8: Cercanía a emociones del clúster «vacunación (2021)»

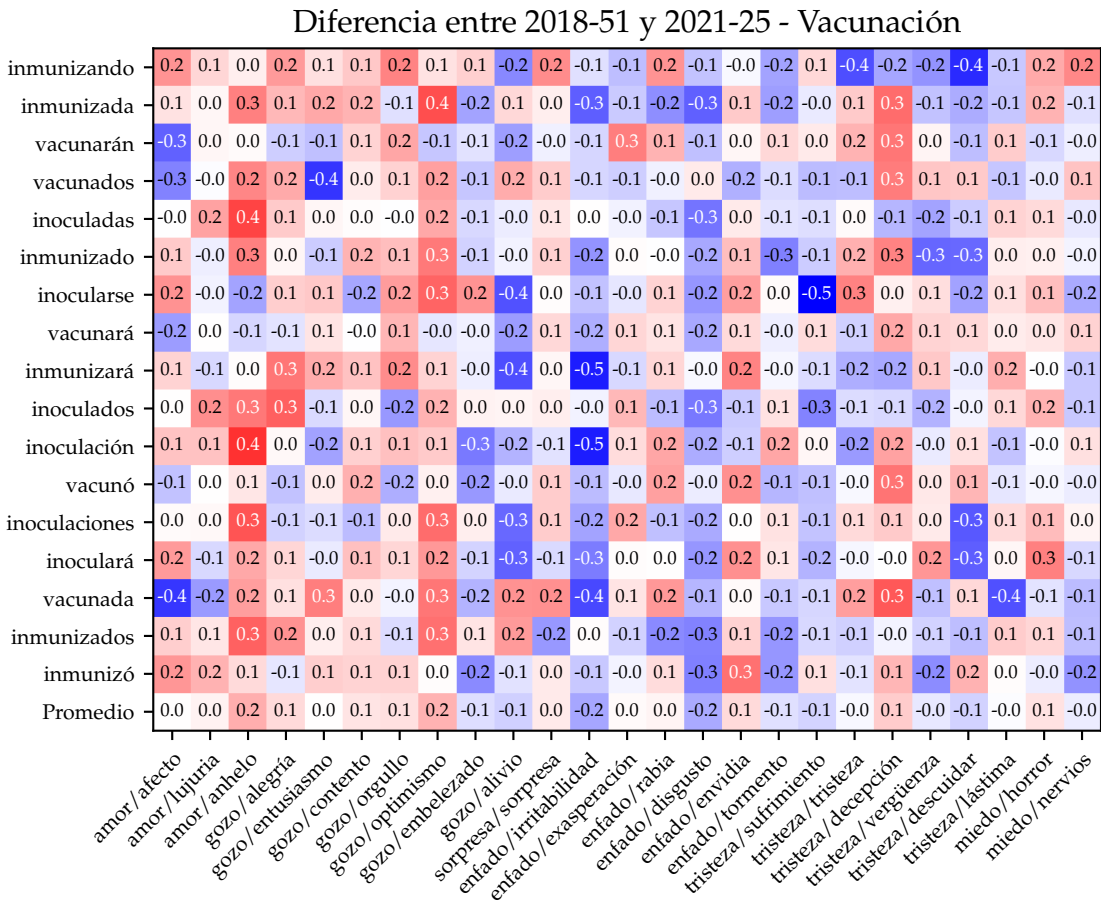

Figura C.9: Cambio en cercanía a emociones del clúster «vacunación»

Apéndice
D

Análisis de desplazamiento semántico utilizando word embeddings diacrónicos del español antes y durante la pandemia de COVID-19

En este apéndice se incluye el artículo «Análisis de desplazamiento semántico utilizando word embeddings diacrónicos del español antes y durante la pandemia de COVID-19». Este fue enviado a consideración a CLEI en el 2021, sin embargo, no fue aceptado.

Análisis de desplazamiento semántico utilizando word embeddings diacrónicos del español antes y durante la pandemia de COVID-19

Esteban Rodríguez Betancourt, **Edgar Casasola Murillo**
Universidad de Costa Rica
estebanrodolfo.rodriguez@ucr.ac.cr, edgar.casasola@ucr.ac.cr

***Abstract*—Words can shift their meaning across time. This case study shows the results obtained by the exploratory analysis of the semantic shifting on Spanish vocabulary using Diachronic Words Embeddings. Diachronic data consists of a 2018 Spanish corpus, created before the COVID-19 outbreak, and a second corpus gathered using a web spider in 2021. We focused on the shifting of specific terms related to different domains such as: affectiveness, political contexts, social contexts and mental health. This paper addresses the construction of the diachronic Spanish word embeddings model as well as the results obtained by the analysis using a non-supervised distance vector technique. The results allowed to identify shifts related to increase of COVID-19 content. Also, we warn about observed shifts explained by bias related to the training data.**

***Index Terms*—Word Embeddings, Diachronic Word Embeddings, Meaning shift**

I. Introducción

Los *word embeddings* se han convertido en una de las principales herramientas para analizar texto en el área de procesamiento de lenguaje natural, al asociar una palabra con una representación vectorial de su «significado» [1]. Estas representaciones pueden ser usadas posteriormente en tareas de recuperación de información, clasificación, generación de texto y otras [2].

Sin embargo, el significado de las palabras es dinámico a través del tiempo [3]. Así que, tiene sentido que las relaciones entre las palabras definidas por los *word embeddings* varíen si un modelo se entrena con corpus recolectados en diferentes momentos. Este tipo de embeddings que se generan utilizando corpus diacrónicos se denominan *embeddings diacrónicos*.

Estos cambios de significado en el tiempo, conocidos como *desplazamiento semántico*, pueden ser visualizados al analizar los cambios en distancia respecto a otras palabras según sus *word embeddings*. Por ejemplo, Hamilton et al. [3] estudiaron los cambios en las palabras en inglés entre el año 1800 y 2009 y en dicho artículo propusieron dos leyes:

1. Ley de conformancia: La tasa de cambios semánticos es proporcional a una potencia negativa de la frecuencia de las palabras.
2. Ley de innovación: Las palabras más polisémicas tienen mayores tasas de cambio semántico.

Por estos cambios en significados se han propuesto modelos de *word embeddings* que consideran la información temporal. Zijun et al. [4] propusieron un método de *word embeddings* condicionado también por la ubicación temporal. Hongyu et al. [5] propusieron un modelo de *embeddings* que fue condicionado por tiempo y lugar (además de la relación entre palabras), con la intención de poder determinar tendencias culturales o situaciones específicas de un lugar.

Esta investigación tiene por objetivo analizar los cambios en las distancias entre palabras, al comparar dos modelos de *word embeddings* entrenados con corpus recolectados en diferentes años (2018 y 2021). Entre los principales aportes de nuestra investigación está la construcción de un corpus diacrónico para el español recolectado desde el internet abierto, orientado al análisis de desplazamiento semántico durante la pandemia de COVID-19. Cabe aclarar que este tipo de estudios son relevantes y su investigación es reciente [6] y es necesario realizar estudios exploratorios en español. Estudios similares han trabajado con corpus específicos sobre COVID-19 en español utilizando *tweets*, pero se han concentrado en generar *embeddings* especializados en COVID-19 [7] y no en estudiar el desplazamiento semántico.

Para la recolección del corpus a partir de la Internet abierta se utilizó una araña. En las siguientes secciones se explicarán los conceptos de *word embeddings* y *crawler* o araña, y se detalla el proceso de obtención del corpus de documentos del 2021 y el entrenamiento del modelo de *word embeddings*.

Posteriormente, se mostrarán diferencias entre ambos modelos: cambios en palabras más cercanas a una palabra y acercamiento o alejamiento de algunas palabras entre los dos modelos. En particular, se revisaron los cambios en términos de afectividad, contexto social, contexto político y salud mental. Además, para algunas palabras se calcularon sus términos más cercanos y otras palabras que más se alejaron y las que más se acercaron.

Finalmente, se presentan recomendaciones para futuros trabajos relacionados con *word embeddings* y posibles dominios de aplicación.

II. Marco teórico

Para la elaboración de este trabajo se usaron diferentes técnicas de recuperación de información y procesamiento de lenguaje natural. En las siguientes secciones, se describe la forma en que se aplicaron dichas técnicas en este trabajo.

$$V[king] - V[man] + V[woman] \sim V[queen]$$

Figura 1. Ejemplo de cómo operaciones aritméticas con *word embeddings* pueden resolver analogías.

Los *word embeddings* son una representación vectorial del «significado» de una palabra. Aunque previamente han existido representaciones vectoriales, como *one-hot encoding*, en este caso consideramos *word embeddings* a las representaciones densas, que reducen el significado a cada palabra a una cantidad fija de dimensiones (por ejemplo, 300 o 100 dimensiones).

Estas representaciones vectoriales presentan propiedades interesantes, que son útiles para hacer análisis de texto natural. Por ejemplo, Bengio et al. [8] describen cómo palabras con significado similar suelen ser cercanas en el espacio vectorial. En el trabajo de Mikolov et al. [2] se muestran ejemplos de resolución de analogías utilizando los *word embeddings*. Este tipo de resolución de analogías se ilustra en la Figura 1.

Existen muchos algoritmos para asociar una palabra a un *word embedding*. Uno de los más conocidos es *Word2Vec* [1] el cual utiliza la capa oculta de una red neuronal para representar cada palabra. Dicha red es entrenada para identificar la palabra oculta dado el contexto (*CBOW*) o el contexto dada la palabra (*Skipgram*). También está GloVe [9], el cual aprovecha la información estadística de las palabras. Otro modelo es *fastText* [10], que entre otras cosas usa información de subpalabras, lo que le permite generar *embeddings* para palabras desconocidas. También hay mejoras en el rendimiento computacional, como BlazingText [11], el cual optimiza *word2vec* y *fastText* para ser calculados en GPU y de forma distribuida.

Los modelos anteriores asocian una palabra a un único vector. Sin embargo, según el contexto las palabras pueden tener significados totalmente diferentes. Existen modelos como *Universal Sentence Encoding* [12] y *Bidirectional Encoder Representations from Transformers* (*BERT*) [13] que son capaces retornar *embeddings* diferentes dependiendo del contexto en el que se utilice la palabra.

Las palabras pueden cambiar su significado a través del tiempo, reflejando cambios tanto en el lenguaje como en la sociedad [14]. Una forma de analizar estos cambios es mediante *word embeddings* diacrónicos [14], los cuales son *word embeddings* generados a partir de un corpus diacrónico. Un corpus diacrónico es un corpus que contiene documentos de varios momentos del tiempo, tal y como lo explica Sierra [15]. En esta investigación se construyeron *word embeddings* utilizando BlazingText [11] con los datos recolectados en marzo 2021 después del inicio de la pandemia por SARS-COV2 o COVID-19, y *embeddings* creados a partir de un corpus recolectado en el 2018 [16]. Tanto el corpus como los *embeddings* construidos como parte de esta investigación están disponibles para uso académico en https://git.ucr.ac.cr/groups/betancourt_casasola_spanish_covid_2021_embedding.

La recolección de datos ser llevó a cabo mediante una araña o *web crawler*. Una araña es un programa que explora la web para obtener documentos. La araña utiliza los hipervínculos en los documentos para encontrar documentos adicionales. Dichos documentos se almacenan para ser usados posteriormente en tareas de recuperación de información (como hacer un buscador web) o bien para entrenar modelos de *machine learning*. Por ejemplo, en este trabajo se utilizó la araña para recolectar documentos que fueron usados para entrenar un modelo de *word embeddings*.

III. Metodología

El proceso de recolección, elaboración de *embeddings* y análisis exploratorio fue el siguiente:

1. Elaboración de un corpus de documentos en español disponibles en el 2021.
   a) Implementación de una araña web.
   b) Recolección del corpus usando la araña web.
2. Elaboración de *word embeddings* usando el corpus generado anteriormente.
3. Comparación entre los dos modelos de word embeddings usados (el generado en este trabajo y el elaborado por Grave et al. [16]).
4. Análisis de cambios en cercanía de las palabras en áreas de afectividad, social, político y salud mental para ciertas palabras seleccionadas.
5. Análisis de cercanía y mayores cambios de similitud para algunas palabras seleccionadas.

A. Generación del corpus de entrenamiento

Para generar el corpus sobre el que se entrenaron los *word embeddings* fue necesario elaborar una araña web. Con dicha araña se obtuvo medio millón de documentos, los cuales se utilizaron para entrenar un modelo de *word embeddings*.

1) Implementación de la araña web: Se implementó una araña web para recolectar un conjunto de documentos de la web. Dicha araña usó una lista de sitios web semilla para iniciar la exploración. De los documentos iniciales se extrajeron los hipervínculos para descubrir más documentos. Este proceso se repitió de forma recursiva hasta llegar al límite deseado de documentos. La araña usó una exploración en anchura primero, para evitar que se limitara a pocos sitios web.

La araña fue implementada en Go[1], para aprovechar sus características de procesamiento paralelo local. Para almacenar los documentos y la cola de tareas se utilizó Badger DB [2]. Para evitar el procesamiento de documentos duplicados se utilizó un filtro bloom en memoria.

La araña lanza varias *gorutinas* (proceso ligero) que consumen URL de la cola de tareas y realizan las siguientes tareas:

Terminación temprana: Se termina el proceso si la araña ya descargó el total de documentos solicitados, se alcanzó la profundidad máxima, o se usó la cantidad reservada de almacenamiento.

[1]Disponible en https://golang.org/

[2]Disponible en https://github.com/dgraph-io/badger

Revisar Robots.txt: Se revisa si el archivo *robots.txt* del sitio permite descargar el documento y si no es así se descarta. El archivo Robots.txt se guarda localmente para futuros usos.

Limitar tasa de descarga Se determina si la araña está descargando datos muy frecuentemente de un mismo dominio. Si es así, el documento se devuelve a la cola para ser procesado posteriormente. Para evitar que la cola retorne constantemente documentos de un mismo dominio estos están ordenados por profundidad (permite anchura primero), hash del URL (aleatoriza el orden en el mismo nivel) y finalmente la URL.

Descargar documento: Se descarga el documento. Si hay redirecciones se descarta la URL usada y se registra la URL nueva. Si hay un exceso de redirecciones la URL se descarta por completo.

Revisar respuesta: Si el estado HTTP es 200 (*Ok*) se continúa con el proceso. Si es 404 (*Not Found*), prohibido (403) o similares se descarta el documento. El documento es reencolado si la respuesta es un error de *Timeout* (408) o *Too Many Requests* (429).

Revisar tipo MIME: Se descartan documentos de tipos no reconocidos. Esta araña solamente acepta documentos de tipo `text/html`.

Parsear documento: Se parsea el documento y se crea un árbol de elementos HTML.

Reemplazar URL por el hipervínculo canónico: Algunos documentos pueden ser accedidos desde múltiples URLs, pero estos pueden declarar una URL como la URL canónica u oficial. En este caso la URL del documento se reemplaza por la declarada en el documento. Esto previene que la araña descargue múltiples versiones del mismo documento.

Extraer vínculos: Se extraen las direcciones apuntadas desde el documento (atributo `href` en la etiqueta `a`), siempre y cuando el vínculo no tenga el atributo «no seguir» (`rel="nofollow"`).

Guardar documentos: Se guarda el documento en una instancia local de BadgerDB y se elimina de la cola de tareas.

2) Generación del corpus: A partir de un conjunto semilla de 22 URL se recolectó un corpus con 500140 documentos. La araña fue configurada para alcanzar una profundidad máxima de 6 niveles, dejar de descargar luego de 500000 documentos y un tamaño máximo en disco de unos 250 GB. La lista de URL iniciales contiene sitios de universidades, prensa y gobierno de Costa Rica, así como prensa internacional:

- https://sites.google.com/presidencia.go.cr/alertas/
- https://www.nacion.com
- https://semanariouniversidad.com/
- https://www.larepublica.net
- https://www.elpais.cr
- https://delfino.cr/
- https://www.ministeriodesalud.go.cr/
- https://www.who.int/es
- https://www.ucr.ac.cr/
- https://www.tec.ac.cr/
- https://www.una.ac.cr/
- https://www.uned.ac.cr/
- https://www.cne.go.cr/
- https://www.ccss.sa.cr/
- https://www.bbc.com/mundo
- https://www.dw.com/es
- https://covid19.go.cr/
- https://www.presidencia.go.cr/
- https://amprensa.com/
- https://www.fedefutbol.com/
- https://www.pulsocr.com/
- https://noticiascostarica.com/

Esta araña fue desplegada entre el 17 de marzo del 2021 y el 18 de marzo del 2021, en una instancia de 16 GB de RAM, 8 núcleos y 320 GB de disco duro SSD en Digital Ocean.

B. Elaboración de los word embeddings

Para calcular los *word embeddings* del corpus recolectado se decidió utilizar BlazingText [11], ejecutado en 4 instancias `ml.c5.9xlarge` en AWS SageMaker Studio y tardó 47:56 minutos. Los parámetros del entrenamiento fueron: `batch_size` 32, `buckets` 10000000, `early_stopping` false, `epochs` 15, `evaluation` false, `learning_rate` 0.05, `max_char` 35, `min_char` 2, `min_count` 5, `min_epochs` 5, `mode` `batch_skipgram`, `negative_samples` 5, `patience` 4, `sampling_threshold` 0.0001, `subwords` false, `vector_dim` 300, `window_size` 5 y `word_ngrams` 2. El preprocesamiento del texto consistió en convertirlo a UTF-8, borrar caracteres inválidos, pasar todo el texto a minúsculas, remover puntuación y normalizar los espacios en blanco a un único espacio (sin saltos de línea).

Los *embeddings* calculados anteriormente usaron un corpus recolectado en marzo del 2021. Para compararlos con un corpus anterior se decidió utilizar los *word embeddings* preentrenados `cc.es.300.vec.gz` disponibles en el sitio web de *fastText* [16]. Estos fueron entrenados sobre textos de *Common Crawl* y Wikipedia.

Ambas colecciones de *embeddings* tienen palabras que no necesariamente están en español. Para limitar las comparaciones a palabras en español se usó el Corpus de Referencia del Español Actual (CREA) de la Real Academia Española [17]. Las comparaciones entre ambos *embeddings* se realizaron con las palabras en común entre ambos *embeddings* y CREA. La cantidad total de palabras se detalla en la Tabla I.

Colección	**|Vocabulario|**
BlazingText sobre corpus descargado en marzo 2021	1.949.099
fastText español 2018 (cc.es.300.vec)	2.000.000
CREA (Abril 2021)	737.799
Intersección de los tres corpus	181.717

Tabla I
TOTAL DE PALABRAS EN CADA COLECCIÓN.

C. Metodología de comparación de los word embeddings

Cada palabra tiene un vector asociado. Las palabras similares tienen un vector similar. Una forma de cuantificar esta similitud es calcular la distancia de cosenos entre los vectores de dos palabras. La distancia de cosenos para dos *word embeddings* se define como:

$$sim(M, W_1, W_2) = \frac{M[W_1] \cdot M[W_2]}{\|M[W_1]\| \cdot \|M[W_2]\|}$$

En la función anterior, M se refiere al modelo de *word embeddings* y $M[W]$ se refiere al *word embedding* de la palabra W.

Distintos modelos de *word embeddings* no son directamente comparables. Incluso un mismo algoritmo con un mismo corpus de entrenamiento podría devolver vectores diferentes para las mismas palabras. Sin embargo, la distancia de cosenos entre pares de palabras suele mantenerse similar. Para comparar el cambio de significado de las palabras entre dos modelos se usó la resta de sus distancias de cosenos:

$$d(M_1, M_2, W_1, W_2) = sim(M_1, W_1, W_2) - sim(M_2, W_1, W_2)$$

Para determinar las palabras más cercanas a una palabra dada se midió la distancia entre esa palabra y el resto de las palabras en el modelo y se tomaron las N palabras con un ángulo menor.

Para encontrar las palabras que más se acercaron o alejaron de una palabra entre dos modelos se tomaron las palabras w con los valores mínimos y máximos al calcular $d(M_1, M_2, W, w) \forall w \in (M_1 \cap M_2 \cap V)$, donde W es la palabra que nos interesa investigar y V es un diccionario de palabras válidas en español. En este caso se usó el Corpus de Referencia del Español Actual de la Real Academia Española [17].

IV. Comparación entre ambos word embeddings

Ambos modelos de *embeddings* fueron construídos sobre documentos recolectados en años diferentes. Los del 2018 (en adelante WE18), fueron recolectados sobre Common Crawl/Wikipedia. Evidentemente, los documentos del 2018 no pueden tener información sobre grandes acontecimientos posteriores, como por ejemplo la pandemia del COVID-19. Estos cambios sí se deberían estar en los *embeddings* construidos sobre un corpus de documentos web recolectado en el 2021 (en adelante WE21).

Al medir los cambios en cercanías entre las 10000 palabras más usadas en español se puede ver que la mayor parte de los cambios fueron cercanos a cero, tal como se muestra en la Figura 2. Esto coincide con la Ley de Conformidad introducida por Hamilton et al. [3].

Un ejemplo de estos cambios está en la palabra «vacuna», que está en la Tabla II. En WE21 las palabras similares son relacionadas específicamente con la vacuna contra el COVID-19, por ejemplo: «anticovid», «tozinamerán»[3]. Incluso se podría decir que «monodosis» está en la lista por la vacuna contra el COVID-19 de una sola dosis desarrollada por Johnson & Johnson.

[3]Nombre comercial de la vacuna contra el COVID-19 desarrollada por Pfizer

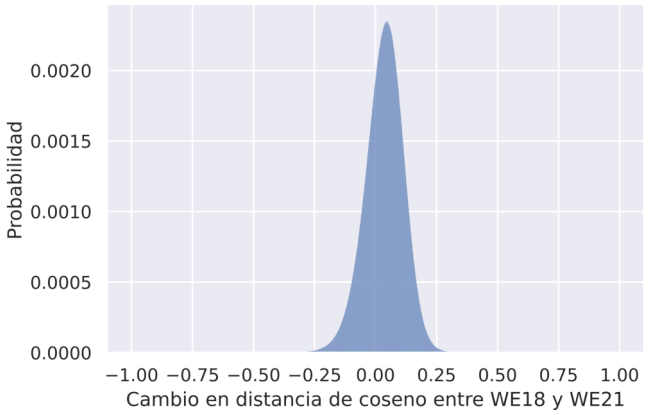

Figura 2. Cambios en similitud entre todos los pares de palabras para las 10000 palabras más usadas que pertenecen a ambos corpus y al Corpus de la Real Academia Española.

WE18	WE21
vacunas	dosis
inmunización	inmunizante
vacunación	aztrazeneca/universidad
Vacuna	inoculación
vacuna.	astrazeneca
autovacuna	anticovid
vacuna-	pfizer
inmunizante	por
vacunara	monodosis
vacunar	tozinamer

Tabla II
PALABRAS MÁS SIMILARES A «VACUNA» SEGÚN WE18 Y WE21.

Por ejemplo, en la Figura 3 se puede ver que «beso», «abrazo» y «saludo» se alejaron entre sí en WE21 respecto a WE18, coincidiendo con la urgencia por practicar el distanciamiento social para prevenir el COVID-19. Otro cambio es «irresponsable», la cual ahora es más cercana a «amistades», «saludo», «beso», «abrazo», «aglomeraciones» y «resfrío». También «fallecer» se acercó a palabras como «amistades», «saludo», «beso», «abrazo», «aglomeraciones» y «resfrío».

V. Cambios en afectividad

Como se muestra en la Figura 4, hubo cambios en afectividad entre los documentos con los que se entrenaron los dos modelos. Por ejemplo, se puede observar que «intensivista» ahora es más cercano a «gratitud» y «agradecimiento», pero también a «resentimiento» o «exhausto», lo que indica que hay más documentos que incluyen ambas palabras de forma cercana. En la dirección contraria, «político» se alejó de palabras como «reconocimiento», «esfuerzo» y «odio».

VI. Cambios en contexto social

En el contexto social, en la Figura 5 «amistad» se alejó más de «amor», «cultura», «solidaridad» y «contacto». También «familia» se alejó de «cultura» e «importancia». Por otro

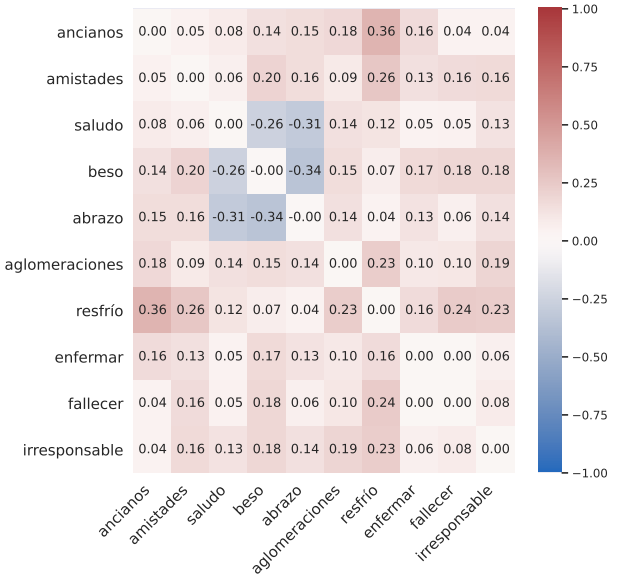

Figura 3. Diferencia en distancia entre algunas palabras entre WE18 y WE21. Valores más altos indican que las palabras se acercaron y menores que se alejaron en WE21 respecto a WE18.

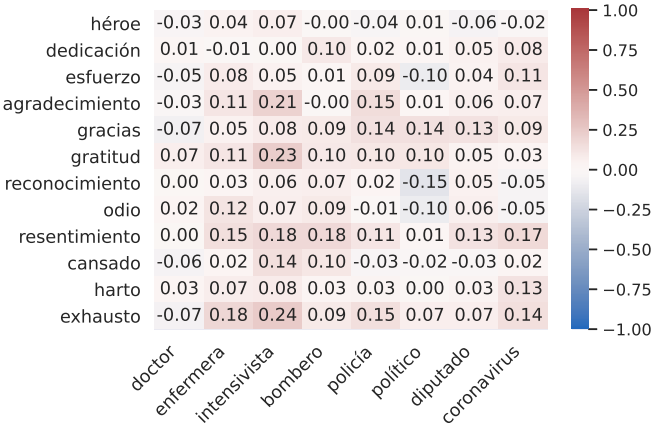

Figura 4. Cambios en afectividad.

lado, «conocidos» se acercó a «cercanía», «importancia», «solidaridad», «trabajo» y «amor».

VII. Cambios en contexto político

Al analizar los cambios en el contexto político se nota parcialización causada por el corpus de entrenamiento. WE18 tiene más información europea o española y WE21 contiene más información latinoamericana, en especial de Costa Rica. Por ejemplo, «diputado» se alejó de «ombudsman» para acercarse a otras como «PLN»[4] o «EPK»[5]. También se ve como «eurodiputado» dejó de ser una de las palabras más cercanas y ahora está «liberacionista» (referente al PLN), nombres de algunos diputados actuales de Costa Rica y adjetivos como «bochornoso».

[4]PLN es Partido Liberación Nacional, un partido político en Costa Rica
[5]EPK es el Partido Comunista de Euskadi

Figura 5. Cambios en contexto social.

Al comparar algunas palabras con puestos de gobierno, en la Figura 6, se puede notar que hubo un acercamiento entre la palabra «ministro» y «salud», lo que indica que dicha cartera ha recibido mayor cobertura en la producción textual (previsiblemente por el COVID-19). Otros acercamientos fueron «bochornoso» con «policía» y «vergüenza» con «presidente».

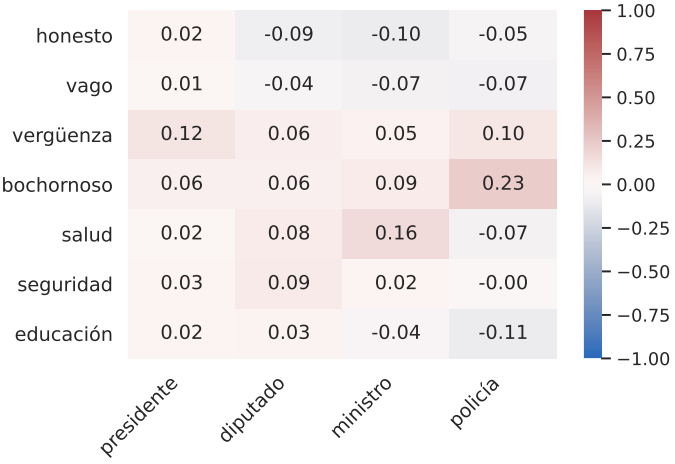

Figura 6. Cambios en el contexto político.

VIII. Cambios en términos de salud mental

En la Figura 7 se muestran los cambios entre entidades y estados de ánimo. «yo» se acercó a todos los estados de ánimo, no así «estoy» que se alejó de «alegre», «feliz», «contento», «entusiasmado», «realizado», «perdido» y se acercó a «depresión», «tristeza», «cansancio», «angustia» y «plenitud». Entre las profesiones del área de la salud hubo un acercamiento a «cansancio», «perdido», «tristeza» pero también a «entusiasmado», «contento» y «plenitud». Estos también se alejaron de «alegre». En el caso de «intensivista» este tuvo cambios más marcados, por ejemplo un acercamiento a «plenitud», «contento», «entusiasmado», «tristeza», «depresión», «angustia», «cansancio» y «estrés».

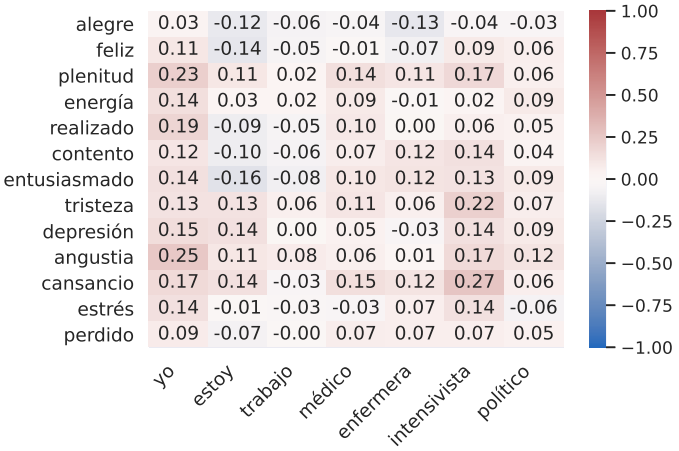

Figura 7. Cambios en términos de salud mental.

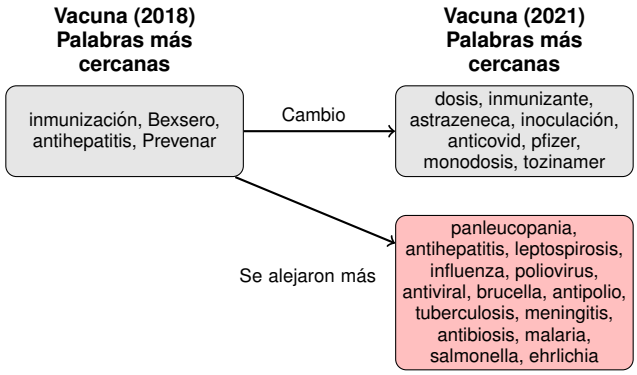

Figura 8. Cambios en las palabras más cercanas y que se alejaron más del término «vacuna».

IX. Análisis de cercanía y mayores cambios

En la Figura 8 se ilustran algunas de las palabras más cercanas al término «vacuna» en el 2018 y en el 2021. Se muestran también las palabras que más se alejaron de vacuna desde el 2018 al 2021. Por otra parte, en la Figura 9 se muestran los términos cercanas al nombre de la empresa Pfizer y las palabras que más se acercaron hacia ese nombre. La Figura 10 muestra los términos cercanos a la palabra «cuarentena» en el 2018 y 2021 y cuáles fueron las palabras que más se alejaron en el 2021 respecto al 2018. En dichas figuras no se muestran las variantes sintácticas de las palabras para mejorar la visualización.

En el caso de «vacuna», esta se alejó de otros tipos de vacuna, como antipolio, tuberculosis, meningitis o malaria y se acercó más a otras palabras como «pausan», «ued[6]» o «cansino[7]».

En el caso de «Pfizer», el *embedding* de esta casa farmacéutica se alejó de varios de sus productos y ahora es más cercana a otras empresas farmacéuticas que fabrican

[6]Unidad de Estancia Diurna [para adultos mayores]. En España han sido centros de vacunación.

[7]CanSino Biologics es una empresa china que está fabricando una de las vacunas contra el COVID-19

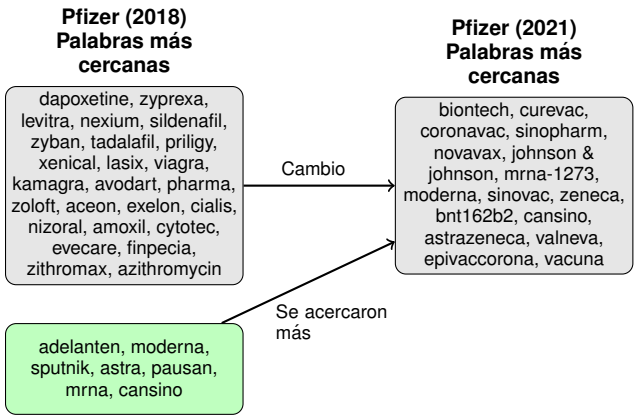

Figura 9. Cambios en las palabras más cercanas y que se acercaron más al término «Pfizer».

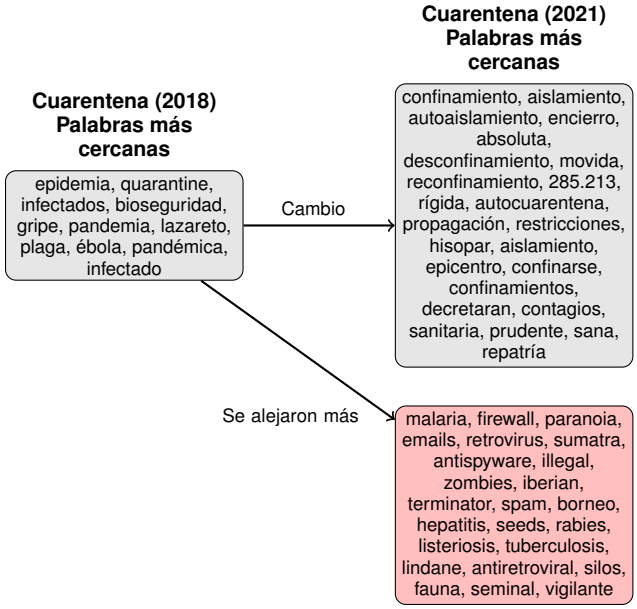

Figura 10. Cambios en las palabras más cercanas y que se alejaron más del término «cuarentena».

vacunas contra el COVID-19 o el nombre de sus vacunas como BioNTech, CureVac, Novavax, Johnson & Johnson, mrna-1273[8]. Algunas de las palabras que más se acercaron fueron Moderna, Sputnik, Astra y Cansino, presumiblemente porque son palabras que adquirieron un nuevo significado que además ganó popularidad.

El término «cuarentena» se alejó más de palabras relacionadas con informática, como firewall, emails, antispyware o spam. «285.213» es una de las palabras más cercanas a la palabra cuarentena en WE21, siendo ese número el total de contagiados de COVID-19 cuando Perú levantó su cuarentena el 30 de junio del 2020.

[8]Nombre clave de la vacuna de Moderna

X. Conclusiones

En este trabajo se mostró la forma en que desarrollamos un *embedding* diacrónico basado en BlazingText para el análisis de desplazamiento simbólico de palabras en español. Nuestro trabajo produce un aporte importante al proveer un nuevo *embedding* diacrónico para realizar este tipo de estudios a la vez que se muestra el tipo de análisis exploratorio desarrollado como caso de estudio.

Mediante la recolección de datos provenientes de la internet abierta se logró observar que del año 2018 al 2021 hubo desplazamientos de significado en vocabulario relacionado con dominios como: afectividad, contexto político, contexto social y de salud mental.

Logramos observar mediante el uso de los *word embeddings* que la posición relativa de las palabras cambió en un periodo corto. En algunas ocasiones se observaron desplazamientos que fueron producto de un sesgo en los datos. Consideramos que esto es apropiado, ya que denota que el nuevo corpus aporta un enriquecimiento semántico regionalizado. Se considera muy importante construir corpus mixtos por región y por área del conocimiento para llevar a cabo este tipo de estudios diacrónicos temporales.

Al desarrollar esta investigación se observó como los movimientos semánticos en algunas ocasiones producen nuevos usos de las palabras que no estaban presentes en los *embeddings* anteriores. Esto debe ser tomado en cuenta por quienes reutilizan *embeddings* siguiendo las recomendaciones de la industria. En este caso es importante balancear el costo de reentrenar un modelo, contra la posible obsolescencia semántica del *embedding* producto de desplazamientos significativos. En el área de investigación, esta situación requiere que se realice investigación ya sea en modelos más baratos de entrenar o bien en modelos que puedan ser entrenados incrementalmente.

Como trabajo futuro, se debe investigar cómo se podrían detectar cambios de significado de las palabras sin necesidad de crear un corpus completo o reentrenar un modelo de *word embeddings* desde cero. Además, es posible que el tratar los *word embeddings* como una serie de tiempo pueda ser beneficioso en tareas de predicción o clasificación. Desde el punto de vista lingüístico este tipo de herramientas se podría utilizar para supervisar en tiempo real la evolución del idioma.

Referencias

[1] T. Mikolov y otros, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [en línea]. Disponible: http://arxiv.org/abs/1301.3781

[2] T. Mikolov, W.-t. Yih, y G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 746–751. [en línea]. Disponible: https://www.aclweb.org/anthology/N13-1090

[3] W. L. Hamilton, J. Leskovec, y D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1489–1501. [en línea]. Disponible: https://www.aclweb.org/anthology/P16-1141

[4] Z. Yao, Y. Sun, W. Ding, N. Rao, y H. Xiong, "Discovery of evolving semantics through dynamic word embedding learning," *CoRR*, vol. abs/1703.00607, 2017. [en línea]. Disponible: http://arxiv.org/abs/1703.00607

[5] H. Gong, S. Bhat, y P. Viswanath, "Enriching word embeddings with temporal and spatial information," *CoRR*, vol. abs/2010.00761, 2020. [en línea]. Disponible: https://arxiv.org/abs/2010.00761

[6] Y. Guo, C. Xypolopoulos, y M. Vazirgiannis, "How covid-19 is changing our language : Detecting semantic shift in twitter word embeddings," 2021.

[7] A. Miranda-Escalada, E. Farré-Maduell, S. L. López, L. Gascó-Sánchez, V. Briva-Iglesias, M. Agüero-Torales, y M. Krallinger, "The profner shared task on automatic recognition of occupation mentions in social media: systems, evaluation, guidelines, embeddings and corpora," in *Proceedings of the Sixth Social Media Mining for Health Applications Workshop & Shared Task*, 2021.

[8] Y. Bengio, R. Ducharme, P. Vincent, y C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003. [en línea]. Disponible: http://www.jmlr.org/papers/v3/bengio03a.html

[9] J. Pennington, R. Socher, y C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1532–1543. [en línea]. Disponible: http://aclweb.org/anthology/D/D14/D14-1162.pdf

[10] A. Joulin y otros, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016. [en línea]. Disponible: http://arxiv.org/abs/1607.01759

[11] S. Gupta y V. Khare, "Blazingtext: Scaling and accelerating word2vec using multiple gpus," in *Proceedings of the Machine Learning on HPC Environments*, ser. MLHPC'17. New York, NY, USA: Association for Computing Machinery, 2017. [en línea]. Disponible: https://doi.org/10.1145/3146347.3146354

[12] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, y R. Kurzweil, "Universal sentence encoder for English," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 169–174. [en línea]. Disponible: https://www.aclweb.org/anthology/D18-2029

[13] J. Devlin, M.-W. Chang, K. Lee, y K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [en línea]. Disponible: https://www.aclweb.org/anthology/N19-1423

[14] A. Kutuzov, L. Øvrelid, T. Szymanski, y E. Velldal, "Diachronic word embeddings and semantic shifts: a survey," in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1384–1397. [en línea]. Disponible: https://www.aclweb.org/anthology/C18-1117

[15] G. Sierra, *Introducción a los corpus lingüísticos*. Universidad Nacional Autónoma de México, Instituto de Ingeniería, 2017.

[16] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, y T. Mikolov, "Learning word vectors for 157 languages," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[17] "Corpus de referencia del español actual," Real Academia Española. [en línea]. Disponible: http://www.rae.es

Bibliografía

Alshahrani, M. (2020). *Exploring embedding vectors for emotion detection* (Tesis Doctoral, University of Essex). Descargado de http://repository.essex.ac.uk/29037/2/Mohammed_Alshahrani.pdf

Alshahrani, M., Samothrakis, S., y Fasli, M. (2017). Word mover's distance for affect detection. En *2017 international conference on the frontiers and advances in data science (fads)* (p. 18-23). doi: 10.1109/FADS.2017.8253186

Asif, M., Zhiyong, D., Iram, A., y Nisar, M. (2020). Linguistic analysis of neologism related to coronavirus (COVID-19). *SSRN Electronic Journal*. Descargado de https://doi.org/10.2139/ssrn.3608585 doi: 10.2139/ssrn.3608585

Balaji, Y. (2021). *Robust learning under distributional shifts.* Digital Repository at the University of Maryland. Descargado de https://drum.lib.umd.edu/handle/1903/27823 doi: 10.13016/P0IH-YX4J

Bengio, Y., Ducharme, R., Vincent, P., y Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155. Descargado de http://www.jmlr.org/papers/v3/bengio03a.html

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*.

Brooke, J., Tofiloski, M., y Taboada, M. (2009, septiembre). Cross-linguistic sentiment analysis: From English to Spanish. En *Proceedings of the international conference RANLP-2009* (pp. 50–54). Borovets, Bulgaria: Association for Computational Linguistics. Descargado de https://aclanthology.org/R09-1010

Butler, C. S., y Simon-Vandenbergen, A.-M. (2021, junio). Social and physical distance/distancing: A corpus-based analysis of recent changes in usage. *Corpus Pragmatics*. Descargado de https://doi.org/10.1007/s41701-021-00107-2 doi: 10.1007/s41701-021-00107-2

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., St. John, R., … Kurzweil, R. (2018, noviembre). Universal sentence encoder for English. En *Proceedings of the 2018 conference on empirical methods in natural language processing: System de-*

monstrations (pp. 169–174). Brussels, Belgium: Association for Computational Linguistics. Descargado de https://www.aclweb.org/anthology/D18-2029 doi: 10.18653/v1/D18-2029

Common Crawl Foundation. (2021). *Common crawl dataset (2018 y 2021).* Descargado de https://commoncrawl.org/

Dean, J., y Ghemawat, S. (2004). MapReduce: Simplified data processing on large clusters. En *Osdi'04: Sixth symposium on operating system design and implementation* (pp. 137–150). San Francisco, CA.

Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2019, junio). BERT: Pre-training of deep bidirectional transformers for language understanding. En *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Descargado de https://www.aclweb.org/anthology/N19-1423 doi: 10.18653/v1/N19-1423

Ekman, P. (1971). Universals and cultural differences in facial expressions of emotion. En *Nebraska symposium on motivation.* Nebraska. Descargado de http://psycnet.apa.org/psycinfo/1973-11154-001

Ester, M., Kriegel, H.-P., Sander, J., y Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. En *Proceedings of the second international conference on knowledge discovery and data mining* (p. 226–231). AAAI Press.

Gong, H., Bhat, S., y Viswanath, P. (2020). Enriching word embeddings with temporal and spatial information. *CoRR, abs/2010.00761*. Descargado de https://arxiv.org/abs/2010.00761

Guo, Y., Xypolopoulos, C., y Vazirgiannis, M. (2021). *How COVID-19 is changing our language : Detecting semantic shift in Twitter word embeddings.*

Gupta, S., y Khare, V. (2017). Blazingtext: Scaling and accelerating word2vec using multiple gpus. En *Proceedings of the machine learning on hpc environments.* New York, NY, USA: Association for Computing Machinery. Descargado de https://doi.org/10.1145/3146347.3146354 doi: 10.1145/3146347.3146354

Hamilton, W. L., Leskovec, J., y Jurafsky, D. (2016, agosto). Diachronic word embeddings reveal statistical laws of semantic change. En *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1489–1501). Berlin, Germany: Association for Computational Linguistics. Descargado de https://www.aclweb.org/anthology/P16-1141 doi:

10.18653/v1/P16-1141

Izsak, P., Berchansky, M., y Levy, O. (2021). *How to train BERT with an academic budget.*

Ji, S., Satish, N., Li, S., y Dubey, P. (2016). *Parallelizing word2vec in multi-core and many-core architectures.*

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., y Grave, E. (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. En *Proceedings of the 2018 conference on empirical methods in natural language processing.*

Joulin, A., Grave, E., Bojanowski, P., y Mikolov, T. (2017, abril). Bag of tricks for efficient text classification. En *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 2, short papers* (pp. 427–431). Valencia, Spain: Association for Computational Linguistics. Descargado de https://aclanthology.org/E17-2068

Jurafsky, D., y Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (1st ed.). USA: Prentice Hall PTR.

Kulkarni, V., Al-Rfou, R., Perozzi, B., y Skiena, S. (2014). Statistically significant detection of linguistic change. *CoRR*, *abs/1411.3315*. Descargado de http://arxiv.org/abs/1411.3315

Kusner, M. J., Sun, Y., Kolkin, N. I., y Weinberger, K. Q. (2015). From word embeddings to document distances. En *Proceedings of the 32nd international conference on international conference on machine learning - volume 37* (p. 957–966). JMLR.org.

Kutuzov, A., Øvrelid, L., Szymanski, T., y Velldal, E. (2018, agosto). Diachronic word embeddings and semantic shifts: a survey. En *Proceedings of the 27th international conference on computational linguistics* (pp. 1384–1397). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Descargado de https://www.aclweb.org/anthology/C18-1117

Levy, O., y Goldberg, Y. (2014, junio). Dependency-based word embeddings. En *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 302–308). Baltimore, Maryland: Association for Computational Linguistics. Descargado de https://aclanthology.org/P14-2050 doi: 10.3115/v1/P14-2050

Li, R., Qin, Z., Wang, X., Chen, S. J., y Metzler, D. (2020). Stabilizing neural search ranking models. En *The web conference 2020 (www).*

Manning, C. D. (2008). *Introduction to information retrieval*. Cambridge University Press. Descargado de https://www.xarg.org/ref/a/0521865719/

Mikolov, T., Chen, K., Corrado, G., y Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR, abs/1301.3781*. Descargado de http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., y Dean, J. (2013). Distributed representations of words and phrases and their compositionality. En *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. proceedings of a meeting held december 5-8, 2013, lake tahoe, nevada, united states.* (pp. 3111–3119). Descargado de http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality

Mikolov, T., Yih, W.-t., y Zweig, G. (2013, junio). Linguistic regularities in continuous space word representations. En *Proceedings of the 2013 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 746–751). Atlanta, Georgia: Association for Computational Linguistics. Descargado de https://www.aclweb.org/anthology/N13-1090

Miranda-Escalada, A., Farré-Maduell, E., López, S. L., Gascó-Sánchez, L., Briva-Iglesias, V., Agüero-Torales, M., y Krallinger, M. (2021). The profner shared task on automatic recognition of occupation mentions in social media: systems, evaluation, guidelines, embeddings and corpora. En *Proceedings of the sixth social media mining for health applications workshop & shared task.*

Montariol, S. (2021). *Models of diachronic semantic change using word embeddings* (Tesis Doctoral, Université Paris-Saclay). Descargado de https://tel.archives-ouvertes.fr/tel-03199801

Navarro-Murillo, N., Calvo-Vargas, P., y Casasola-Murillo, E. (2019). Identification of unsuitable content for children in video gaming forums. En *2019 IV jornadas costarricenses de investigación en computación e informática (JoCICI)* (p. 1-6). doi: 10.1109/JoCICI48395.2019.9105201

Papakyriakopoulos, O., Hegelich, S., Serrano, J. C. M., y Marco, F. (2020). Bias in word embeddings. En *Proceedings of the 2020 conference on fairness, accountability, and transparency* (p. 446–457). New York, NY, USA: Association for Computing Machinery. Descargado de https://doi.org/10.1145/3351095.3372843 doi: 10.1145/3351095.3372843

Pennington, J., Socher, R., y Manning, C. D. (2014). Glove: Global vectors for word representation. En *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, october 25-29, 2014, doha, qatar, A meeting of sigdat, a special interest group of the ACL* (pp. 1532–1543). Descargado de

http://aclweb.org/anthology/D/D14/D14-1162.pdf

Real Academia Española. (2021). *Corpus de referencia del español actual.* http://www.rae.es. Real Academia Española. Descargado de http://www.rae.es

Rehurek, R., y Sojka, P. (2011). Gensim—-python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, *3*(2).

Ren, F., y Liu, N. (2018, 04). Emotion computing using word mover's distance features based on Ren_CECps. *PLOS ONE*, *13*(4), 1-17. Descargado de https://doi.org/10.1371/journal.pone.0194136 doi: 10.1371/journal.pone.0194136

Schönemann, P. H. (1966). A generalized solution of the orthogonal Procrustes problem. *j-PSYCHO*, *31*(1), 1–10.

Sharir, O., Peleg, B., y Shoham, Y. (2020). The cost of training NLP models: A concise overview. *ArXiv*, *abs/2004.08900*.

Shaver, P., Schwartz, J., Kirson, D., y O'Connor, C. (1987). Emotion knowledge: Further exploration of a prototype approach. *Journal of Personality and Social Psychology*, *52*(6), 1061–1086. Descargado de https://doi.org/10.1037/0022-3514.52.6.1061 doi: 10.1037/0022-3514.52.6.1061

Sierra, G. (2017). *Introducción a los corpus lingüísticos*. Universidad Nacional Autónoma de México, Instituto de Ingeniería.

Yao, Z., Sun, Y., Ding, W., Rao, N., y Xiong, H. (2017). Discovery of evolving semantics through dynamic word embedding learning. *CoRR*, *abs/1703.00607*. Descargado de http://arxiv.org/abs/1703.00607

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., y Stoica, I. (2010). Spark: Cluster computing with working sets. En *Proceedings of the 2nd usenix conference on hot topics in cloud computing* (p. 10). USA: USENIX Association.