UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

CREACIÓN DE UN MODELO DE BÚSQUEDA LÉXICA UTILIZANDO TÉCNICAS BASADAS EN GRAFOS

TESIS

QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
JORGE CARLOS REYES MAGAÑA

TUTOR:
DR. GERARDO EUGENIO SIERRA MARTÍNEZ
INSTITUTO DE INGENIERÍA

COMITÉ TUTOR:
DRA. GEMMA BEL ENGUIX
INSTITUTO DE INGENIERÍA
DR. GIBRÁN FUENTES PINEDA
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

CIUDAD UNIVERSITARIA, CD. MX., JUNIO 2021

A Celi y mi nené (†)
A mis papás, Luis y Betty

Agradecimientos

A Dios, por las bendiciones en mi vida.

A mi esposa Celi, tanto lo vivido y en tan poco tiempo. Gracias por ser mi acompañante durante este pequeño paso de nuestra existencia.

A mis papás Luis y Betty, mis hermanos Luis Felipe, Erick Ricardo y Daniel Jesús, por estar siempre. Tenerlos como familia es de los más grandes tesoros que Dios me pudo regalar.

A mis amigos de Mérida y la Ciudad de México, que me apoyaron para seguir adelante.

A la Universidad Nacional Autónoma de México, por abrirme las puertas una vez más y permitirme seguir creciendo académicamente.

A la Universidad Autónoma de Yucatán y el Programa para el Desarrollo Profesional Docente, que me brindaron el apoyo para poder realizar los estudios de Doctorado.

Al Dr. Gerardo Sierra Martínez, por presentarme el mundo del Procesamiento de Lenguaje Natural y por todas sus enseñanzas que durante 4 años ampliaron mi conocimiento en el ámbito de la investigación.

A la Dra. Gemma Bel Enguix, por forjar mis capacidades que lograron un pequeño avance en la lexicografía computacional, así como fomentar mi crecimiento profesional, sobre todo en el área de investigación.

Al Dr. Gibrán Fuentes Pineda por la retroalimentación que me dio durante mis evaluaciones semestrales, que sin duda ayudaron a mejorar mi trabajo de investigación.

A la Dra. Helena Gómez Adorno, porque con su ejemplo y dedicación a la investigación, me incentivan a trabajar más por el desarrollo de la ciencia.

A la Dra. Darnes Vilariño Ayala, por el tiempo otorgado durante las etapas de la tesis.

A Lulú, Amalia y Cecilia, gracias por el apoyo que siempre me brindaron frente a todas las dudas administrativas permitiendo que el proceso sea más ligero.

A todos mis compañeros del Grupo de Ingeniería Lingüística, que a través de diferentes generaciones me acompañaron en los estudios doctorales, desde su trabajo y dedicación me motivaron a seguir adelante.

Índice general

Índice de tablas

Índice de figuras

Creación de un modelo de búsqueda léxica utilizando técnicas basadas en grafos

by

Jorge Carlos Reyes Magaña

Abstract

The development of inverse dictionaries over the years has been approached from various perspectives, from the production of specialized books for this purpose, to the development of technological applications that solve this task. This research work addresses a methodology for the creation of a lexical search model to be used in the implementation This research work addresses a methodology for the creation of a lexical search model to be used in the implementation of this type of dictionary. The selected search engine uses technology based on graph theory, whose nodes and edges are based on a corpus of word association norms. A small corpus of non-formal definitions was used to evaluate the dictionary. The results showed that the performance of the method is better than other current inverse dictionaries, outperforming various natural language processing techniques, ranging from information retrieval models to the use of innovative methods with vectorial word representations.

Creación de un modelo de búsqueda léxica utilizando técnicas basadas en grafos

por

Jorge Carlos Reyes Magaña

Resumen

El desarrollo de diccionarios inversos a través de los años ha sido abordado desde diversas perspectivas, desde la fabricación de libros especializados para este fin, hasta el desarrollo de aplicaciones tecnológicas que resuelven esta tarea. Este trabajo de investigación aborda una metodología para la creación de un modelo de búsqueda léxica que se utiliza en la implementación de un diccionario de este tipo. El motor de búsqueda seleccionado utiliza tecnología fundamentada en teoría de grafos, cuyos nodos y aristas están basados en un corpus de normas de asociación de palabras. Para la evaluación del diccionario se utilizó un pequeño corpus de definiciones no formales. Los resultados mostraron que el desempeño del método es mejor que otros diccionarios inversos actuales, superando diversas técnicas de procesamiento de lenguaje natural, ya sea desde modelos de recuperación de información hasta el uso de métodos innovadores con representaciones vectoriales de palabras.

Capítulo 1

Introducción

1.1. Planteamiento del problema

Existen dos tipos de diccionarios para poder ligar un concepto con su significado: semasiológico y onomasiológico. El primero proporciona significados, es decir, dada una palabra, el usuario obtiene el significado de esa palabra. El segundo funciona en la forma contraria, dada la descripción de una palabra, el usuario obtiene el concepto relacionado [Baldinger, 1970].

Los diccionarios semasiológicos impresos son de uso común y su funcionamiento es sencillo, simplemente se busca en un índice la palabra deseada y después de una consulta rápida se llega a su definición. En el caso de los diccionarios onomasiológicos impresos, son menos frecuentes, un exponente importante dentro de este ámbito, es el Tesauro Roget, escrito en inglés en el siglo XIX. En este tipo de diccionarios, la composición es diferente, las palabras se encuentran organizadas por la semántica compartida o características asociadas agrupadas bajo palabras claves [Sierra, 2000b; Sierra y McNaught, 2003], de esta forma al navegar por la estructura del diccionario se llega a la palabra objetivo. Una desventaja es que su actualización es complicada, dada su naturaleza física; agregar nuevas palabras cada cierto tiempo, puede resultar no ideal. Aunado a esto, el funcionamiento del diccionario es tedioso y, en caso de no tener buenas pistas que guíen adecuadamente a la palabra objetivo, limitan en gran medida este tipo de recursos. Es así, que la creación de soluciones de tipo onomasiológicas plantean un área de investigación

en la que se centra este trabajo.

El advenimiento de la tecnología, la inteligencia artificial y el auge de la lingüística computacional, permite visualizar que es posible resolver el problema desde estas perspectivas. Se han incorporado nuevas aproximaciones a diccionarios onomasiológicos, también conocidos como diccionarios inversos; las versiones realizadas permiten encontrar una palabra objetivo dada su definición en lenguaje natural y devuelven el listado de palabras posibles. Los diccionarios inversos en línea, en su mayoría se presentan en inglés, un ejemplo es *OneLook* [1], sin embargo su versión en español[2] es bastante limitada al no proporcionar resultados adecuados. Algunas herramientas, como los buscadores de Internet, se podrían considerar como elementos auxiliares para encontrar palabras dada su definición; sin embargo, al no ser desarrollados con este objetivo específico, algunas veces pueden traer información no relevante.

El problema que plantean las búsquedas onomasiológicas también ha sido abordado desde otra perspectiva. Para la psicología, y especialmente la psicolingüística, el problema es formulado como de acceso léxico. Las contribuciones más importantes son de Zock y Biemann [2020], Bonin [2004], Levelt [2005], Aitchison [2012] y Jarema *et al.* [2002]. La principal discusión en el área es cómo tratar las latencias y errores en el acceso, el fenómeno de la *punta de la lengua*. En años recientes, ha habido la clara necesidad de interacción entre la lingüística, psicología y tecnologías del lenguaje para poder combatir algunas enfermedades relacionadas a la disnomia. La línea de investigación relacionada a la cognición motivó la inclusión de la tarea relacionada al acceso léxico[3] en la producción del lenguaje para el *Cognitive Aspects of the Lexicon (CogaLex) Workshop* en la *Conference on Computational Linguistics*, 2014.

Por parte de la psicolingüística se han desarrollado conjuntos de datos, llamados normas de asociación de palabras, que permiten conocer las palabras y sus relaciones, esto se realiza mediante pruebas en las que participan voluntarios para recolectar las asociaciones. Algunos ejemplos de estas colecciones son, para el español, las Normas del Español de México [Arias-Trejo *et al.*, 2015]; y para el inglés, las normas de *Edinburgh Associative Thesaurus* [Kiss *et al.*,

[1] `https://www.onelook.com/reverse-dictionary.shtml`

[2] `https://rimar.io/`

[3] `http://pageperso.lif.univ-mrs.fr/~michael.zock/ColingWorkshops/CogALex-4/cogalex-webpage/pst.html`

1973] y las *South Florida Free Association Norms* [Nelson *et al.*, 1998]. Las normas puede ser tratadas a través de la teoría de grafos dada su naturaleza y estructura para su posterior procesamiento. La teoría de grafos ha sido de gran impacto en el desarrollo del procesamiento de lenguaje natural, se han implementado métodos para detectar plagio, resúmenes y clasificación de textos, entre otros [Gómez-Adorno *et al.*, 2016; Pinto *et al.*, 2014; Vilariño *et al.*, 2013]. En este trabajo de investigación, la idea es utilizar grafos, tomando en cuenta la estructura que presentan las normas de asociación, y posteriormente encontrar palabras objetivo.

Con base en la descripción del problema a tratar, se presenta la siguiente hipótesis.

1.2. Hipótesis de investigación

La estructura que presentan las normas de asociación de palabras, dan paso a un procesamiento que puede ser abordado con algoritmos de carácter matemático, en particular, por medio de algoritmos de grafos, permitiendo así la creación de diccionarios inversos digitales que realizan el proceso de localización de términos basado en la centralidad de los nodos, obteniendo de manera precisa las palabras que se están buscando.

1.3. Objetivo

Construir un diccionario inverso digital, basado en algoritmos de centralidad de grafos y en un corpus de normas de asociación de palabras, que sea capaz de obtener la palabra más precisa, con base en una definición no formal, en formato escrito y realizada en lenguaje natural.

1.3.1. Objetivos específicos

- Identificar y comparar qué algoritmos basados en grafos permiten realizar una localización eficiente y eficaz de un concepto basado en un consulta en lenguaje natural. La diversidad de algoritmos existentes en la teoría de grafos es amplia, sin embargo, al ser un aporte nuevo al área de lexicografía computacional, es necesario descubrir cuales de estos algoritmos clásicos son los más adecuados para resolver una búsqueda léxica.

- Explorar el corpus más adecuado para la construcción del grafo sobre el que se realizarán búsquedas. La importancia de este corpus radica en que una correcta implementación del grafo es un pilar fundamental de la búsqueda léxica, sobre la que se ejecutarán los algoritmos basados en grafos.

- Construir herramientas que coadyuven en la resolución de problemas de procesamiento de lenguaje natural aplicadas. La presente investigación, aborda de manera principal una problemática referente al área de lexicografía computacional. Sin embargo, el tipo de soluciones existentes para esta área, la mayoría de las veces se enfocan al desarrollo de tecnologías para diccionarios semasiológicos, dejando descubierta el área de los onomasiológicos. Existe esta brecha que debe ser cubierta con herramientas innovadoras, un ejemplo de ello es el desarrollo de *word embeddings*, elementos que han sido de gran ayuda para el desarrollo de PLN.

1.4. Organización de la tesis

Con base en la descripción general de esta investigación y su desarrollo, se presenta la siguiente organización de la tesis.

En el capítulo 1 se presenta el planteamiento del problema, la hipótesis de investigación, así como los objetivos alcanzables durante el desarrollo del trabajo.

En el capítulo 2 se describen las bases teóricas relacionadas con el acceso léxico y el problema de la punta de la lengua.

Posteriormente, en el capítulo 3 se revisan las búsquedas onomasiológicas existentes en la actualidad, pasando desde elementos físicos (como libros), hasta los diccionarios inversos que son implementados con diversas tecnologías de inteligencia artificial.

En el capítulo 4 se identifican las formas computacionales y tipos de estructuras, usadas para el procesamiento de textos en el Lenguaje Natural. Después, en el capítulo 5 se muestran los diversos modelos de búsqueda léxica que fueron implementados para la creación del diccionario inverso, con las adecuaciones pertinentes en cada caso de acuerdo a cada algoritmo de

centralidad.

El capítulo 6 expone los experimentos realizados, los resultados obtenidos en cada modelo probado, así como la evaluación del sistema. La comparación de todos los algoritmos implementados permite identificar cuál de ellos fue mejor para la búsqueda léxica. También se presentan algunos resultados que fueron obtenidos al trabajar con normas de asociación, permitiendo generar estructuras vectoriales como resultado paralelo de esta investigación.

Finalmente, se desarrollan las conclusiones en el capítulo 7, así como trabajos futuros.

En el apéndice A se muestra el corpus de definiciones que se utilizó durante la experimentación del modelo desarrollado, tanto para español como inglés.

Capítulo 2

Acceso léxico

Los diccionarios inversos permiten la obtención de un concepto basado en su descripción y ayudan a resolver el problema de la punta de la lengua, que se presenta cuando no es posible recordar adecuadamente un término que se escapa de la memoria, pero existe una idea general de su descripción o incluso ciertos elementos que ayudan a definirlo. Generalmente se le identifica también como un problema de acceso léxico. El acceso léxico forma parte de la cognición en los seres humanos. El estudio de estos procesos ha sido importante para psicólogos y expertos del área.

Para la realización de este tipo de recursos han existido diferentes aproximaciones, desde diccionarios físicos hasta plataformas digitales y cada uno de ellos con sus propias ventajas y desventajas. Esta tesis presenta una metodología para la creación de un diccionario inverso, la cual está basada en técnicas de grafos y permite realizar consultas en lenguaje natural. El modelo es simple y eficiente en la obtención de las palabras objetivo.

Este capítulo inicia con una introducción de los aspectos más relevantes en el desarrollo de la cognición, donde uno de los elementos fundamentales es el surgimiento del lenguaje. Posteriormente, se abordan los aspectos más notorios del problema de la punta de la lengua. Finalmente, se presentan las normas de asociación de palabras que establecen un mapeo de las estructura mental de las palabras en el cerebro. Este tipo de recursos psicolingüísticos son de suma importancia, ya que establecen un eje primordial en la presentación del modelo de

búsqueda léxica.

2.1. Cognición

Davis [2014] menciona que la palabra cognición tiene sus origenes en el latín, donde el vocablo *cognoscere* significa "llegar a conocer", refiriéndose de manera específica a la "acción y efecto de conocer". La cognición también está relacionada con la *conciencia* y el *criterio*, es decir, con el conocimiento que el ser humano tiene de su propia existencia, de sus estados y de sus actos; por su parte, el *criterio* establece una serie de reglas a la cual se establece una determinación.Tanto la cognición como el aprendizaje mantienen una relación estrecha. Durante el desarrollo de una persona a lo largo de su vida, se van adquiriendo todo tipo de experiencias que le permiten forjar apreciaciones nuevas, así como la formulación de nuevos conceptos. El aprendizaje basado en la experiencia se genera cuando el conocimiento recién adquirido provoca un cambio en el comportamiento.

2.2. Niveles del proceso cognitivo

De acuerdo con Ruiz *et al.* [2016], la psicología cognitiva tiene cabida como una subdisciplina de la psicología. Su principal aportación se refiere al estudio de los procesos que se relacionan con la generación de conocimiento, abordados desde un sentido general, es decir, desde la forma de percibir las cosas, el almacenamiento de la información, en cómo se realiza el aprendizaje, la manera de razonar y establecer la atención sobre elementos particulares. El proceso de comunicarse también es considerado como un elemento destacado de la psicología cognitiva. Estos mismos autores establecen que algunos procesos de interés para los psicólogos cognitivos son los siguientes:

- **Percepción.** Permite establecer la manera de recibir información y su agrupamiento para identificar lo que esta representa. Esta información, que llega a través de sensores, es combinada junto con el conocimiento previamente adquirido, con la finalidad de hacerlo entendible. En resumen, es la interpretación de la información recibida.

- **Memoria.** Permite identificar la forma de codificar, almacenar y recuperar la información recibida. Elementos importantes para este proceso son las memorias tanto de corto y largo plazo, memoria semántica, memoria cotidiana y la memoria episódica, a través de sus diferentes interrelaciones. En este proceso podemos encontrar enfermedades asociadas a la memoria, como lo son: Alzheimer, demencias, amnesias, etc.

- **Atención.** Este proceso establece que para poder realizar una tarea es necesario enfocar las capacidades que poseen los individuos. Algunos ejemplos son la atención sostenida y los diversos factores que la provocan por medio de señales o motivaciones, así como la atención selectiva.

- **Razonamiento.** Es uno de los procesos más complejos de la cognición. Primeramente, es una actividad mental que permite vincular el procesamiento con el entendimiento de la información, para que al relacionar los sucesos se realicen acciones acordes con lo que está sucediendo. Provoca en los individuos la ejecución de procesos de generalización, predicción, etc.

- **Lenguaje.** Establece la forma en la que los seres humanos se comunican, la mayoría de las veces a través del habla, pero no es la única forma. Se crea un sistema de elementos discretos que por medio de reglas se combinan en una infinidad de oraciones.

- **Toma de decisiones.** Se refiere al proceso de adoptar creencias, afirmaciones o realizar acciones de una serie de opciones. Existen dos tipos de decisiones, las racionales e irracionales.

- **Motivación.** Este proceso es el motor que guía a los individuos a perseguir una meta, que a su vez la influye y permite su perseverancia. Con este proceso se activan las funciones emocionales y de carácter cognitivo, que conducen las acciones de forma, ya sea deliberada o premeditada hacia un propósito específico.

- **Emoción.** Dentro del ámbito de la psicología cognitiva, este proceso no era considerado importante . A pesar de ello, el estudio de las respuestas emocionales, visto desde un

enfoque cognitivo a través del procesamiento de información, resultó de gran utilidad para su entendimiento.

2.3. Lingüística cognitiva

De acuerdo con Fajardo Uribe [2007], una característica fundamental en los seres humanos que los hace distintivos es el uso de la razón. Debido a esto, a la razón se le ha considerado objeto de estudio en el ámbito filosófico durante muchos años. Algunos procesos como la resolución de problemas, evaluación de situaciones, comprensión de uno mismo y de las demás personas, comprensión del mundo e inferencias, son resultado del uso de la razón. Esta autora menciona que el lenguaje es el actor principal de la conformación de conceptos, de la abstracción y la forma de entender el mundo. Asimismo, se le considera como el elemento que permite conectar la razón y la mente con el pensamiento. Actualmente se han desarrollado diferentes escuelas y visiones referentes al lenguaje en relación a los procesos cognitivos. En este contexto, nace la lingüística cognitiva, involucrando múltiples disciplinas para estudiar el fenómeno lingüístico. Debido a la naturaleza del lenguaje, desde sus inicios como ciencia, la lingüística ha trabajado de manera paralela a otras ciencias. Las bases del lenguaje que se manifiesta en los diversos grupos sociales, tienen un fundamento neurológico y psicológico, que trabajan con referencia a elementos de la lógica formal. Estas combinaciones entre disciplinas dieron origen a la antropolingüística, neurolingüística, sociolingüística, etnolingüística, psicolingüística y lingüística matemática, entre otras.

El desarrollo de las ciencias y la tecnología de los últimos siglos ha permitido el crecimiento exponencial de información cuyo principal medio de transmisión ha sido el ambiente tecnológico y en diferentes formatos, uno de los cuales, y de principal interés para el desarrollo de esta tesis, es el de textos digitales, abriendo un basto campo de estudio a través de herramientas computacionales que permitan procesar de manera automática el volumen de información disponible hoy en día.

Con base en lo anterior, surge la lingüística computacional como campo interdisciplinar que

involucra modelos computacionales asociados al lenguaje natural, que permiten a las computadoras el entendimiento de los idiomas humanos. Algunas veces el término Procesamiento del Lenguaje Natural (NLP en inglés, *Natural Language Processing*) es usado para referirse a este tipo de de tecnologías [Sidorov, 2001].

2.4. El problema de la punta de la lengua

La producción de lenguaje, de forma oral o escrita, consiste de manera general en escoger palabras que implícitamente se pueden recuperar. Para ello, simplemente se confía en la información almacenada en el cerebro y cuando se presentan problemas en el uso del vocabulario, se le puede preguntar a otra persona, o utilizar un recurso externo (diccionario) [Zock y Biemann, 2020]. Esta producción de lenguaje se realiza de manera automática, la mayoría de las veces accediendo en tiempo real al bagaje de palabras con la que los individuos cuentan en su mente. Sin embargo, algunas veces cuando las palabras que se necesitan para expresarse resultan inaccesibles de manera inmediata, es necesario encontrar mecanismos que ayuden a resolver el problema. Es de reconocer que este tipo de problemas no se dan siempre, y por lo general se dan en escenarios específicos.

El problema del acceso léxico es abordado también como el problema de la punta de la lengua, *tip-of-the-tongue problem, ToT* por sus siglas en inglés, que ha sido tratado por diversos autores desde una perspectiva piscológica, estudiando los diversos mecanismos cognitivos relacionados con este problema [Brown, 1991, 2012; Nickels, 2000]. Estas investigaciones han merecido estudios por parte de expertos a lo largo de los años.

De acuerdo con Zock y Biemann [2020], el problema de la punta de lengua se presenta cuando una persona conoce lo que quiere decir, conoce la forma correspondiente, pero por alguna razón no puede acceder a ella en el momento que la necesita, ya sea de manera hablada o escrita. Los autores establecen que algunas de las posibles razones para ello son las siguientes:

- Proximidad al nivel semántico o fonológico.
- Baja frecuencia de la palabra objetivo.

- Ausencia de uso.
- Edad, ya sea por la pérdida de neuronas o destrucción de las conexiones entre ellas.
- Distracción.
- Interferencia.

Para ayudar a las personas a resolver el problema de la punta de lengua, existen diversas herramientas, incluyendo desde diccionarios hasta sistemas computacionales que hacen uso de la tecnología aplicada a procesamiento del lenguaje natural. Cuando el acceso léxico se realiza a través de un diccionario, es necesario que éste proporcione mecanismos adecuados y eficaces para poder obtener la palabra objetivo. Un diccionario semasiológico resulta no ser adecuado cuando se intenta recuperar una palabra de la que únicamente se posee la descripción de sus características; esto es debido principalmente a que la estructura del diccionario se basa en un índice alfabético y, al hacer una búsqueda recorriendo todo el contenido, resultaría en una tarea poco eficiente y seguramente sin un resultado satisfactorio. Por ello, los diccionarios onomasiológicos resultan un recurso más adecuado en esta búsqueda; la descripción de los principales diccionarios que caen en esta categoría se describen en el siguiente capítulo.

El problema de la punta de la lengua o de acceso léxico, se traduce finalmente como un problema de búsqueda, como indican Zock *et al.* [2010], fomentando el desarrollo de nuevos enfoques interdisciplinarios al problema. El acceso léxico fue el tema de la tarea en el workshop Cogalex en Coling 2014. En esta competencia varias estrategias nuevas fueron presentadas para poder resolver el problema. Ghosh *et al.* [2014] introdujeron un modelo de dos etapas que demostró ser muy eficiente.

En esta tesis, se presenta el diseño de un modelo de búsqueda léxica que permite encontrar palabras objetivo basadas en consultas hechas con lenguaje natural. Se utilizan como espacio de búsqueda para el modelo, un corpus de normas de asociación de palabras, que traducido en una estructura de tipo grafo, permite mediante algoritmos de centralidad, encontrar una palabra objetivo. Bajo esta misma idea, se transfirió la metodología pero ahora aplicada al idioma inglés, mostrando que el uso de corpus de normas de asociación en conjunto con la teoría de grafos,

resulta una sinergia positiva para la obtención exitosa de palabras objetivo durante el problema del acceso léxico.

2.5. Normas de asociación de palabras

Las asociaciones libres de palabras se construyen de manera común cuando se presenta una palabra estímulo a un participante, y se le pide que produzca de manera verbal o escrita la primera palabra que se le venga a la mente. La contestación obtenida es llamada la palabra respuesta. Es importante señalar que estas asociaciones al estímulo no vienen clasificadas ni etiquetadas de ninguna forma, es decir que no se almacenan las relaciones semánticas entre los estímulos y sus repuestas, como sinonimia, hiperonimia, hiponimia, etc. Sin embargo, esta relacionalidad entre el par estímulo-respuesta está presente de manera implícita, ya sea perteneciendo a la misma categoría, co-ocurriendo en el mismo contexto, o por medio de una relación de causa-efecto, etcétera.

Las compilaciones de estas asociaciones libres de palabras son llamadas Normas de Asociación de Palabras. Muchas lenguas tienen este tipo de recursos. Para su construcción es necesario de mucho tiempo, así como muchos voluntarios. Entre las compilaciones existentes, las más conocidas en inglés son las *Edinburgh Associative Thesaurus*[1] (EAT) [Kiss *et al.*, 1973] y la colección la Universidad del Sur de Florida[2] [Nelson *et al.*, 1998].

El único recurso diseñado y compilado utilizando técnicas clásicas, para el Español Mexicano, es el Corpus de Normas de Asociación de Palabras para el Español de México[3] (NAP) [Arias-Trejo *et al.*, 2015].

La metodología de esta tesis, propone el uso de las normas de asociación de palabras, como la base del diseño de un sistema de búsqueda léxica que trabaja desde las pistas o definiciones al concepto, es decir, desde las respuestas al estímulo. El español es el idioma seleccionado para la mayor parte de los experimentos y sus correspondientes resultados. Sin embargo, para definir el alcance de esta metodología, se muestra la transferibilidad a otros idiomas y por lo tanto

[1] `http://rali.iro.umontreal.ca/rali/?q=en/Textual%20Resources/EAT`
[2] `http://web.usf.edu/FreeAssociation`
[3] `http://www.labpsicolinguistica.psicol.unam.mx/Base/php/general.php`

algunas de las pruebas también se realizaron en el idioma inglés.

2.5.1. Normas de asociación de palabras en español

Para español, existen diversos corpus de palabras de asociación. Algarabel *et al.* [1998] integra 16,000 palabras, incluyendo análisis estadísticos de los resultados. Macizo *et al.* [2000] construye normas para 58 palabras en niños, y Fernández *et al.* [2004] trabaja con 247 elementos léxicos que corresponden al español [Sanfeliu y Fernandez, 1996]. Las normas anteriores no están disponibles para su análisis y uso en los experimentos para probar la metodología. Sin embargo, las siguientes normas, que se describen con mayor amplitud, sí están disponibles para su procesamiento y uso.

2.5.1.1. Normas de asociación de palabras para el español de México

El corpus NAP de Arias-Trejo *et al.* [2015] fue elaborado con una muestra de 578 adultos jóvenes, hombres (239) y mujeres (339), con un rango de edad que va desde los 18 a los 28 años, y con un rango de educación de al menos 11 años. El número total de tokens del corpus es 65731, con 4704 palabras diferentes.

Para esta tarea se usaron 234 palabras estímulo, todas ellas sustantivos comunes tomados del Inventario de Compresión y Producción de palabras MacArthur [Jackson-Maldonado *et al.*, 2003]. Es importante mencionar que, si bien los estímulos son siempre sustantivos, las palabras asociadas son de selección libre, es decir, los informantes pueden relacionar a la palabra estímulo con cualquier palabra sin importar su categoría gramatical.

En la tabla 2-1 se presenta un ejemplo de las respuestas que se obtuvieron aplicadas al estímulo **agua**; por cuestiones de espacio y legibilidad solo se presentan las primeras diez respuestas. Adicionalmente a las respuestas obtenidas, existen valores numéricos asociadas a cada una de ellas. En este caso, NAP en su recolección reportó 3 valores.

Los *estímulos* se dividieron en dos listas A y B de 117 palabras cada una. Para cada *estímulo* y sus respuestas, los autores investigaron diferentes medidas, descritas de la siguiente manera:

Tiempo. Mide los segundos que el participante tarda en dar una respuesta para cada *estímulo*.

Respuesta	Frecuencia	Tiempo	% Asociación
sed	36	3.71	13.0909
beber	24	3.04	8.7273
tomar	18	3.43	6.5455
vida	18	3.53	6.5455
fresca	16	3.76	5.8182
fría	12	3.67	4.3636
manguera	12	4.44	4.3636
limpia	11	4.57	4.0000
líquido	8	4.60	2.9091
azul	7	3.92	2.5455

Tabla 2-1: Primeras 10 respuestas del estímulo **agua** para NAP

Frecuencia. Establece el número de ocurrencias de cada una de las palabras asociadas a un *estímulo*.

Fuerza de asociación. Relaciona la frecuencia con el número de respuestas para cada estímulo. Se calcula de la siguiente manera: siendo FP la frecuencia de una palabra determinada asociada a un *estímulo*, y ΣF la suma de las frecuencias de las palabras conectadas el mismo *estímulo* (el número total de respuestas), la fuerza de asociación (FA) de la palabra W a dicho *estímulo* se obtiene con la fórmula:

$$FA_W = \frac{FP * 100}{\Sigma F}$$

2.5.1.2. Normas de Asociación Libre en Castellano

Las Normas de Asociación Libre en Castellano (NALC) de Fernández *et al.* [2010] incluyen 5,819 palabras estímulo y sus correspondientes respuestas obtenidas a partir de las respuestas de asociación libre. El número total de palabras diferentes es 31,207.

En el estudio de normas de asociación participaron 1500 estudiantes universitarios que tenían el español como lengua nativa y participaron de manera voluntaria en el estudio empírico.

Los criterios bajos los cuales se construyeron las NALC contienen respuestas para:

- 247 palabras que, según el estudio normativo de Sanfeliu y Fernandez [1996], corresponden a las denominaciones más frecuentes en castellano de los dibujos estandarizados de Snodgrass y Vanderwart. Aunque las normas de los dibujos recogen la denominación más frecuente para un total de 254 estímulos pictóricos, en este trabajo se han utilizado solamente 247 nombres. El conjunto de estímulos utilizados constaba de 238 palabras que eran nombres simples y 9 términos que eran nombres compuestos.

- 664 palabras extraídas de las respuestas de las 55 palabras presentes en las normas de falso recuerdo de Fernández *et al.* [2015].

- 1,372 palabras extraídas en su mayoría del estudio normativo de González [1996].

- 3,536 palabras extraídas de diversos estudios normativos en castellano y de las respuestas de asociación de estas mismas normas.

El procedimiento reportado indica que la recogida de respuestas se llevó a cabo en sesiones de grupo de aproximadamente una hora de duración, en las que el número de sujetos oscilaba entre 20 y 40. En cada una de las sesiones se distribuyeron aleatoriamente ejemplares de los cuadernillos de respuesta. Los sujetos realizaron la tarea de manera individualizada y a su propio ritmo.

Como es habitual en este tipo de estudios, en las instrucciones se pedía al sujeto que leyera, una por una, todas las palabras impresas en las páginas del cuadernillo, y que al lado de cada una de estas palabras escribiera otra palabra, "la primera palabra en la que pienses después de leer la palabra impresa". Se ponía énfasis en que ante cada estímulo escribieran lo primero que les viniese a la cabeza y que lo hicieran tan rápido como les fuera posible.

La tabla 2-2 presenta las primeras diez respuestas, obtenidas con NALC, aplicadas al estímulo **agua**. En este caso, el valor numérico hace referencia a la proporción de cada respuesta respecto al total de ellas aplicadas al mismo estímulo, también conocido como fuerza de asociación.

Respuesta	Fuerza de asociación
sed	0.135
beber	0.075
limpia	0.055
transparente	0.045
grifo	0.035
líquido	0.035
vaso	0.035
cristalina	0.03
fresca	0.03
mar	0.03

Tabla 2-2: Primeras 10 respuestas del estímulo **agua** para NALC

2.5.2. Normas de asociación de palabras en inglés

En esta sección se presentan las dos principales normas de asociación de palabras del idioma inglés, ambas correspondientes a finales del siglo XX. La colección comprende una variedad de estímulos, cada una de ellas con más de 5000.

2.5.2.1. Edinburgh Associative Thesaurus (EAT)

Las normas de asociación de Edimburgo [Kiss *et al.*, 1973] fueron recolectadas tomando como base un conjunto nuclear de palabras y posteriormente se fueron agregando elementos a la colección. Las respuestas fueron recolectadas respecto a las palabras del conjunto nuclear y después estas mismas palabras fueron usadas como estímulos para obtener sus futuras respuestas y así sucesivamente. El conjunto nuclear fue obtenido de:

- Los 200 estímulos usados en las normas de Palermo y Jenkins [1964].
- Las mil palabras más frecuentes de Thorndike y Lorge [1944].
- El vocabulario básico de Inglés de Ogden [1930].

El proceso se repitió tres veces hasta que el número de las respuestas diferentes era tan

grande que no pudieron ser reusadas como estímulos. La recolección de datos partió del conjunto nuclear y se detuvo cuando 8,400 estímulos se habían usado. Todas las respuestas dadas en inglés o unidades verbales fueron incluidas, incluso algunas frases y numerales. Los datos recogidos cubrieron un rango de formas gramaticales y formas no flexionadas.

Cada estímulo fue presentado a 100 personas diferentes, cada una de las cuales recibió 100 palabras. Esto dio lugar a un total de 55,732 palabras en la colección. Cada persona recibió una hoja impresa con los 100 estímulos en orden aleatorio, con el fin de minimizar el efecto de sesgo, que se da cuando la presencia de ciertos estímulos influye en las respuestas posteriores. En las instrucciones dadas a los sujetos se les pedía que escribieran la primera palabra presente en su pensamiento, trabajando lo más rápido posible. El tiempo total en realizar este proceso fue medido, y la mayoría de los sujetos completaron la hoja en un rango de 5 a 10 minutos.

Los sujetos fueron en su mayoría estudiantes de licenciatura de una variedad de universidades británicas. El rango de edades de los sujetos fue de 17 a 22 años con una moda de 19. La distribución de género fue de 64 % hombres y de 36 % mujeres. Los datos fueron recolectados desde junio de 1968 hasta mayo de 1971, en su mayoría fueron obtenidos en un salón bajo supervisión. Las hojas que tuvieron más de un 25 % de respuestas vacías fueron desechadas.

La tabla 2-3 presenta las primeras diez respuestas al estímulo *water*, en este caso se presentan dos columnas numéricas que refieren a la frecuencia y a la proporción de las respuestas respecto al total.

2.5.2.2. South Florida Free Association Norms

Las normas de asociación libre de la Universidad del Sur de Florida [Nelson *et al.*, 1998] son la base de datos de asociaciones libres más grande que ha sido recolectada en los Estados Unidos de América. Más de 6,000 participantes produjeron cerca de 750,000 respuestas a 5,019 palabras estímulo. A los participantes se les pidió que escribieran la primera palabra que tuvieran en mente que fuera significativamente relacionada o fuertemente asociada a la palabra presentada. Los formatos dados a los participantes mostraban las palabras estímulo y a su lado un espacio en blanco para ser llenado. Por ejemplo, si se les presentaba *WATER* _____, debían

Respuesta	**Frecuencia**	**Fuerza de asociación**
wet	13	0.13
drink	8	0.08
tap	6	0.06
sea	5	0.05
cold	3	0.03
hot	3	0.03
h2o	3	0.03
rain	3	0.03
river	3	0.03
thirst	3	0.03

Tabla 2-3: Primeras 10 respuestas del estímulo ***water*** para *EAT*

escribir la palabra asociada en el espacio designado para ello. A este proceso se le denomina asociación discreta, porque a cada participante se le pide únicamente una respuesta por cada estímulo.

Las normas empezaron a recolectarse en 1973. El proceso de obtención se realizó con un promedio de 149 (D.S. 15) participantes, para ello se les presentaron de 100 a 120 palabras en inglés en un folleto, conteniendo de 20 a 30 palabras por página con un orden aleatorio dentro de todo el folleto. La gran mayoría de las palabras de estas normas son: sustantivos con 76 %, adjetivos con 13 % y verbos con 7 %. Algunas otras partes de la oración también fueron representadas. Adicionalmente, un 16 % fueron identificados como homógrafos. Los autores de estas normas de asociación señalan que para la selección de las palabras estímulos, se identificaron aquellas que habían sido estudiadas en varios experimentos de memoria. Algunas otras fueron seleccionadas porque fueron producidas como respuestas en las normas de rimas de Nelson y McEvoy [1979]. Algunas de ellas fueron añadidas por trabajos realizados en el área de primación [Bajo, 1988; Cañs, 1990; McEvoy, 1988; Nelson *et al.*, 1991]. Adicionalmente, para estudiar el problema de la concretización [Nelson y Schreiber, 1992]. Muchas palabras, concretas o abstractas, fueron tomadas de normas de concretización [Paivio *et al.*, 1968; Toglia y Battig, 1978].

La tabla 2-4 presenta las primeras diez respuestas al estímulo *water*, la tabla presenta dos columnas numéricas, la primera de ellas es la frecuencia y la segunda representa la proporción de las respuestas respecto al total.

Respuesta	**Frecuencia**	**Fuerza de asociación**
drink	23	0.16
cool	11	0.07
wet	10	0.07
swim	8	0.05
thirsty	7	0.05
pool	6	0.04
faucet	5	0.03
thirst	5	0.03
ice	4	0.02
cold	3	0.02

Tabla 2-4: Primeras 10 respuestas del estímulo ***water*** para *South Florida Free Association Norms*

2.5.3. Métodos actuales para la creación de normas de asociación

En años recientes, la web se ha vuelto la forma natural de obtener datos para construir tales recursos. *Jeux de Mots*[4] proporciona un ejemplo en francés [Lafourcade, 2007], mientras que *Small World of Words*[5] contiene colecciones en 14 lenguas, en el momento de escritura de esta tesis. Dichos repositorios tienen el problema de que son recolectados sin control sobre los participantes, la habilidad lingüística de los usuarios, su edad, género o nivel de estudios.

El uso de normas libres de palabras para procesar relacionalidad entre palabras no es nuevo. Borge-Holthoefer y Arenas [2009] describen un modelo *Random Inheritance Model (RIM)* para extraer relaciones de similitud semántica usando información de asociaciones libres. Los autores aplican metodologías de redes para descubrir vectores de características. Los vectores obtenidos

[4]http://www.jeuxdemots.org/
[5]https://smallworldofwords.org/.

fueron comparados con representaciones vectoriales basados en LSA(*Latent Semantic Analysis*) y modelos WAS(*Word Association Space*).

Bel-Enguix *et al.* [2014] usaron técnicas de análisis de grafos para calcular asociaciones de grandes colecciones de textos. Adicionalmente, Garimella *et al.* [2017] publicó un modelo de asociaciones de palabras que era sensible al contexto demográfico. Esto estaba basado en una arquitectura de redes neuronales con *n-gramas* de saltos de palabras. Este método mejoró el rendimiento de los métodos genéricos para calcular asociaciones que no toman en cuenta la demografía de los participantes. Sinopalnikova y Smrz [2004] presentaron un marco de trabajo metodológico para la construcción y extensión de redes semánticas con un tesauro de asociación de palabras, *word association thesaurus (WAT)*, incluyendo una comparación de la calidad e información proporcionada por WAT y otros recursos de lenguaje. Los autores mostraron que WAT es comparable con corpus balanceados y pueden ser utilizados en sustitución de ellos, en caso de ausencia del corpus.

De Deyne *et al.* [2016] introdujeron un modelo de activación expansivo con la finalidad de codificar la estructura semántica de una red de palabras de asociación, específicamente una parte de *Small World of Words*. El modelo basado en asociaciones fue comparado con un modelo de embeddings *(word2vec)* usando parámetros que miden relacionalidad y similitud entre palabras, obteniendo un promedio de mejora del 13 % comparado con *word2vec*.

Roth y Im Walde [2008] utilizaron tres recursos lingüísticos: un diccionario, entradas de *Wikipedia* y un corpus de co-ocurrencias para obtener un conjunto de datos que proporcionen información de relacionalidad semántica. Las pruebas realizadas para validar su metodología fueron desarrolladas utilizando un conjunto de asociados (verbos y sustantivos) en alemán. Encontraron que los recursos lingüísticos en conjunto se complementan de manera adecuada para obtener información semántica de calidad.

2.5.3.1. Creación automática de normas de asociación de palabras para el español de México

Anteriormente se describieron las únicas normas de asociación de palabras para el español de México, NAP, que contiene únicamente 234 estímulos. Una cantidad reducida de palabras, y con la finalidad de ampliar normas para el español de México, Reyes-Magaña *et al.* [2020] propusieron una metodología independiente del lenguaje que permite obtener este tipo de recursos que se denominan Normas Automáticas de Asociación de Palabras, NAAP en lo sucesivo.

La metodología propuesta necesita de dos elementos fundamentales: un diccionario y una conjunto de vectores pre-entrenados. De manera específica se usó el Diccionario del Español de México [DEM, 2010] y para los vectores se tomaron aquellos disponibles en Español, disponibles en un repositorio de vectores[6]. Los algoritmos usados para el entrenamiento de estos vectores fueron: FastText [Bojanowski *et al.*, 2017], Word2Vec [Mikolov *et al.*, 2013], y Glove [Pennington *et al.*, 2014]. Los detalles de estas tecnologías serán explicados en el capítulo 4.

El proceso general consiste en recorrer el diccionario entero, trabajando con las entradas y sus definiciones. Se consideró que todas las entradas toman el papel de palabras estímulo, y cada una de las palabras que definen la entrada se vuelven las respuestas. El proceso también establece la inferencia de un valor numérico que mide la relación entre las palabras, esto permite obtener un peso equivalente al valor que tienen las normas de asociación clásicas.

Algoritmo 1: Normas Automáticas de Asociación de Palabras

```
   Datos: Diccionario, Vectores de palabras
   Resultado: NAAP
1  pre-procesar(Diccionario)
   para cada entrada en diccionario hacer
2      para cada palabra en definición hacer
3          similitud = similitud_coseno(entrada,palabra);
4          peso = similitud * tf_idf(palabras);
5      fin
6      ordenar(palabras)
7  fin
```

[6] https://github.com/dccuchile/spanish-word-embeddings

El algoritmo 1 presenta el esquema general del modelo. El Diccionario del Español de México DEM [2010] es el resultado de investigaciones del vocabulario usado en México desde 1921. Es un diccionario de corte descriptivo, tomando criterios lingüísticos en su elaboración.

Algunas veces las definiciones de cada una de las entradas cuentan con ejemplos de uso. Esta información fue eliminada ya que agregaba información no relevante en las NAAPs finales. Por ejemplo, la definición de "taco" proporciona la variedad de tacos existentes en México, como son: taco de tripa, taco de cabeza, o incluso taco de ojo, tomándolo en su sentido metafórico; todos estos ejemplos resultan poco probables de tener como respuestas en una NAP clásica.

Por lo tanto, con la finalidad de preparar las definiciones se realizaron algunos pasos de procesamiento, como son:

- Todas las palabras fueron lematizadas usando Freeling [Padró y Stanilovsky, 2012]
- Se eliminaron las palabras funcionales, para ello se tomaron las palabras vacías del conjunto disponible en el paquete de *NLTK* [Bird y Loper, 2004].
- Se agregaron algunas palabras muy particulares al conjunto de palabras vacías, con la finalidad de removerlas también. Estas palabras son muy comunes en los diccionarios por su naturaleza lexicográfica y no proporcionan datos significativos a las NAAP. Algunas de ellas son: "aproximadamente", "generalmente", "específicamente", "tipo", "etcétera", entre otras.

Después de haber removido las palabras descritas en los puntos anteriores, se trabajó con las palabras restantes, calculando la similitud coseno entre cada entrada y cada palabra correspondiente a la definición. En este punto se utilizaron los vectores de palabras pre-entrenados para poder hacer las operaciones matemáticas correspondientes. La tabla 2-5 presenta las características principales para cada modelo vectorial.

Los corpus utilizados para entrenar estos vectores son los siguientes: FastText, Glove y Word2Vec con un corpus de billón de palabras en español o SBWC (*Spanish Billion Word Corpus*) y FastWiki entrenado con Wikipedia en español.

Nombre corto	**Modelo**	**Dimención**	**# vectores**	**Algoritmo**
FastText	FastText con SBWC	300	855,380	FastText con Skipgram
Glove	GloVe con SBWC	300	855,380	GloVe
Word2Vec	Word2Vec con SBWC	300	1,000,653	Word2Vec con Skipgram
FastWiki	FastText con Wikipedia en Español	300	985,667	FastText con Skipgram

Tabla 2-5: Características de los vectores pre-entrenados en Español.

De la misma manera que en el paso anterior, se tomaron estas palabras restantes para calcular el valor *tf-idf*, en los que cada definición es tratada como un documento por separado. El valor se usó como factor de ajuste en la similitud coseno de las palabras. El peso es calculado usando la siguiente fórmula:

$$P_{fa}(estímulo, respuesta) = tf\_idf(respuesta) * similitud\_coseno(estímulo, respuesta) \quad (2\text{-}1)$$

A este peso se le llamó Fuerza de Aproximación (P_{fa}). El paso final consiste en ordenar de mayor a menor los pesos de todas las respuestas asociadas (palabras en una definición) a una entrada (estímulo).

El corpus de las NAAP está disponible en Github[7]. Se generaron 4 colecciones diferentes, una para cada conjunto de vectores pre-entrenados usados en el experimento durante la medición de la similitud coseno. La distribución de los archivos se encuentra en formato *Excel* y la organización alfabética de todas las palabras que conforman el recurso es visible en cada una de las pestañas del documento.

En las tablas 2-6, 2-7, 2-8 y 2-9 se aprecian las primeras diez respuestas obtenidas usando esta metodología, correspondientes al estímulo “agua”. Se observa una diversidad en el orden, la cual es debida a la similitud coseno medida a partir de cada modelo de vectores presentando en la tabla 2-5.

[7] `https://github.com/jocarema/AWAN`

Respuesta	**Fuerza de aproximación**
líquido	0.644
lluvia	0.497
evaporar	0.471
lago	0.455
hielo	0.416
río	0.405
calentamiento	0.385
mar	0.365
olor	0.361
sabor	0.347

Tabla 2-6: Primeras 10 respuestas del estímulo agua usando FastText.

La metodología descrita anteriormente, pretende inferir relaciones semánticas entre los estímulos y sus respuestas. Las principales relaciones mostradas en NAP son: metonimia, meronimia, funcionalidad, cohiponimia, hiponimia, "hecho de" y sinonimia [Mijangos *et al.*, 2017]. El objetivo de las NAAP es capturar las relaciones semánticas pero no los tipos de relación.

Para la evaluación de las NAAP, se realizaron dos tipos de pruebas:

- **Intrínseca.** Mide qué tanto las palabras obtenidas con esta metodología son representativas. Esta comparación se realiza a través de vectores, que a su vez se obtienen de las NAP en español con metodologías clásicas. Para este fin, se utilizaron los conjuntos de datos: *MC-30* y *WordSim-353*. Ambos fueron creados utilizando anotadores humanos y sirvieron como punto de referencia para la comparación.

- **Extrínseca.** Se ha demostrado que las NAP son buenas en la recuperación de conceptos de un diccionario inverso [Reyes-Magaña *et al.*, 2019a]. Esta prueba permite observar el comportamiento de las NAAP en esta tarea.

Las pruebas demostraron que las NAAP obtenidas son competitivas en los dos tipos de tareas. Los resultados de estos dos tipos de pruebas se muestran en el capítulo 6.

Respuesta	**Fuerza de aproximación**
evaporar	0.59
líquido	0.554
lluvia	0.516
calentamiento	0.403
lago	0.392
olor	0.387
precipitar	0.377
hielo	0.372
río	0.367
sabor	0.353

Tabla 2-7: Primeras 10 respuestas del estímulo agua usando FastWiki.

Respuesta	**Fuerza de aproximación**
líquido	0.618
lluvia	0.558
mar	0.532
río	0.499
lago	0.472
hielo	0.464
vida	0.445
olor	0.401
calentamiento	0.39
sabor	0.368

Tabla 2-8: Primeras 10 respuestas del estímulo agua usando Glove.

Respuesta	**Fuerza de aproximación**
líquido	0.63
evaporar	0.531
río	0.518
lluvia	0.507
mar	0.486
lago	0.478
hielo	0.448
calentamiento	0.374
olor	0.359
nube	0.329

Tabla 2-9: Primeras 10 respuestas del estímulo agua usando Word2Vec.

Capítulo 3

Búsqueda onomasiológica

Después de haber abordado de manera general el problema de acceso léxico, es importante estudiar los principales aportes que se han desarrollado para resolver este problema. En este capítulo se señalan dos destacadas vertientes, la primera trata sobre aproximaciones de carácter físico, a través principalmente de libros especializados (diccionarios) para tal fin, algunos incluso pertenecientes a finales del siglo XIX. Posteriormente, se hace un estudio de los aportes hechos basados en las nuevas tecnologías en PLN. Desde esta perspectiva, la tarea de acceso léxico se puede abordar como una tarea de búsqueda de información (*retrieval information*); a pesar de que esta búsqueda se basa en la localización de documentos que contengan información específica, es posible comparar esta tarea con la búsqueda de un concepto asociado a su descripción. También se estudian otras aproximaciones basadas en modelos de vectores de palabras que, al tomar ventaja de los últimos avances en PLN, permiten determinar si su aplicación es adecuada para la resolución del problema de acceso léxico.

3.1. Diccionarios semasiológicos y onomasiológicos

La lexicografía, según la Real Academia Española, es parte de la lingüística que estudia los principios teóricos en que se basa la composición de diccionarios[1]. Los diccionarios ayudan

[1] `https://dle.rae.es/lexicografía`

a entender el significado de las palabras; su localización generalmente se hace a través de un orden alfabético. Este tipo de diccionarios de uso común, según la lexicografía, son clasificados como diccionarios semasiológicos. La semasiología parte de una palabra individual y busca la información semántica asociada a esa palabra, básicamente responde a la pregunta ¿cuál es el significado de esa palabra? [Geeraerts, 2003].

Por otra parte, los diccionarios onomasiológicos funcionan de manera opuesta que los semasiológicos. Parten del significado y proveen un rango de opciones léxicas que expresan esa noción, es decir, van del significado a la palabra [Hartmann, 2005]. En la siguiente sección se aborda de manera más amplia este tipo de diccionarios.

Ullman [1983] utiliza tres elementos para definir el concepto de significado: nombre, sentido y objeto. *Nombre* es el sonido que especifica la palabra, *sentido* es toda la información que el nombre le comunica al oyente y *objeto* es la característica o evento no lingüístico del que se habla. Siendo así, que el significado es la relación que existe entre el nombre y su sentido. De acuerdo con Ullman, esta relación se puede dar en diversas direcciones, ya que varios nombres pueden estar asociados al mismo sentido (sinonimia) y varios sentidos al mismo nombre (polisemia). Además, explicita que esta relación entre nombre y sentido es "recíproca y reversible", indicando que se puede pasar de una a otra. De esta forma, cuando las personas se comunican a través del lenguaje se está produciendo un proceso de significación y designación.

Desde la perspectiva semasiológica, la significación es un proceso que parte del nombre hacia el sentido. El oyente recibe sonidos y, desde las significaciones en su mente, obtiene el significado. En sentido inverso, la perspectiva onomasiológica, se parte del sentido y se dirige al nombre. Un hablante utiliza designaciones pertenecientes a su vocabulario para expresar objetos mentales.

Baldinger y Wright [1980] establecen que la semasiología y la onomasiología son enfoques opuestos. Los autores utilizan el triángulo semántico de Ullman para presentar las relaciones estructurales entre un objeto o realidad, un sentido o concepto, y un nombre o imagen acústica, como se puede apreciar en la Fig. 3.1.

El diccionario semasiológico es un elemento auxiliar en la decodificación, esto permite tomar el papel del hablante: parte desde la forma de expresión hasta llegar a la significación,

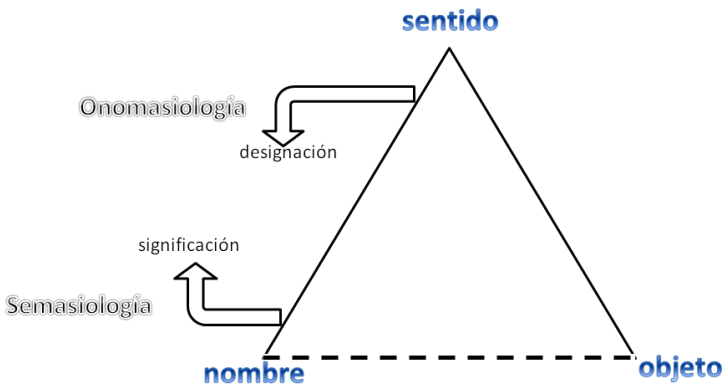

Figura 3-1: Relación dual entre onomasiología y semasiología [Baldinger y Wright, 1980].

estableciendo significados con expresiones; de esta forma la disposición en las entradas permite relacionar una palabra con su significado. En contraposición, los diccionarios onomasiológicos se presentan como ayuda en la codificación. El papel del hablante ahora se realiza partiendo del objeto mental para buscar designaciones. La disposición transita desde el significado hasta el nombre.

Para el uso de los diccionarios onomasiológicos es necesaria una consulta adecuada, con la finalidad de obtener los mejores resultados, necesitando un bagaje de vocabulario, que generalmente se enriquece de experiencias sociales, de carácter cultural y son dependientes de antecedentes geográficos de las personas. Es así, que la descripción de los conceptos puede consistir en aproximaciones desde múltiples perspectivas. De aquí que el desarrollo de un diccionario onomasiológico debe considerar la mayoría de características para establecer un concepto, así como las diferentes formas de nombrarlo. Posteriormente, se debe interpretar de manera adecuada esta descripción para devolver exitosamente la palabra que se está buscando.

3.2. Tipos de diccionarios de búsqueda onomasiológica

Diversos autores indican que los usuarios mas comunes de este tipo de diccionarios son los escritores. Sin embargo, su uso no es exclusivo de ellos. Cada tipo de usuario tiene diferentes necesidades. Algunos de los usos más importantes son:

- Identificar el sinónimo apropiado. Durante la escritura de textos, algunas veces los autores conocen una palabra, pero quieren encontrar el sinónimo más adecuado, tomados de una variedad que muestran diferencias sutiles y su vez comparten algunos elementos. Mas allá de entender una palabra que no se conoce, lo que se pretende es encontrar una palabra de la que ya se sabe el significado, pero asociando nexos léxicos.

- Acrecentar el léxico. Es decir, se desea clarificar una idea que se desea expresar, pero el vocabulario del autor no comunica lo que desea. Existen dos formas para lograr esto, la primera es usar las palabras que tienen a su alcance como posibles opciones, y segundo obtener nuevas palabras que le permitan aumentar su vocabulario.

- Solucionar el problema de la punta de la lengua. Como mencionamos en el capítulo 2, este problema se presenta cuando una persona tiene la necesidad de expresar una palabra, pero por diversas razones no le es posible encontrarla y usarla. Lo más frustrante de estas situaciones, es que los usuarios saben que la palabra sí existe, pero simplemente se les olvida.

- Identificar una palabra para un significado específico. Esto generalmente sucede cuando se usa un idioma extranjero o cuando se están usando términos muy técnicos. En estos casos, el usuario desconoce si la palabra realmente existe. Esta situación se da en actividades de traducción e interpretación asociadas al significado de las palabras.

- Estudiar un idioma. Los procesos de aprendizaje de lenguas comienzan conociendo las palabras en el idioma del estudiante, pero desconociendo su traducción en la lengua destino. Con este tipo de obras, se pretende que los usuarios puedan llegar a las palabras que permitan expresar las ideas que se tienen en la mente.

3.2.1. Diccionarios onomasiológicos impresos

Cuando se habla de diccionarios onomasiológicos, se hace referencia a todos aquellos diccionarios usados con la finalidad de encontrar una palabra partiendo de una idea. En este tipo de obras, que se destacan principalmente en su forma de organizar las palabras, ya que no se

encuentran solitarias; se presentan agrupadas de acuerdo con propiedades compartidas y agrupadas bajo palabras clave. De acuerdo con Sierra [2000b], se reconocen cuatro tipos de libros según el tipo de información contenida, la estructura y el tipo de búsqueda realizada: tesauros, diccionarios inversos, diccionarios de sinónimos y antónimos, y diccionarios pictóricos.

A pesar de que estas obras se pueden encontrar en formato electrónico, la forma de estructurarlas es idéntica, así que no hay diferencia en la descripción ni en la forma de buscar las palabras.

3.2.1.1. Tesauros

Son considerados como los diccionarios onomasiológicos de mayor antigüedad, por ejemplo, el *Onomasticon* de Julius Pollux [1706]. La localización de la palabras parte del significado hacia el significante. La clasificación presentada en estas obras utiliza jerarquías temáticas, que dependen principalmente de cada autor que las realiza. Las personas que consultan este tipo de obras fluyen entre las temáticas que los llevarán a las palabras que están relacionadas con un concepto. En lo referente a la lexicografía moderna, una de las obras más importantes es el *Thesaurus of English Words and Phrases* (1852) de Roget. Esta obra está disponible para el idioma inglés y presenta una clasificación temática con 6 clases: *abstract relations, space, matter, intellectual faculties, voluntary powers,* y *sentient and moral powers*, que a su vez se expanden en 39 secciones y 990 cabeceras. Siguiendo esta estructura se encuentra el Diccionario de ideas afines y elementos de tecnología [Benot *et al.*, 1895], obra escrita en español. El Diccionario ideológico de la lengua española [Casares, 1942] presenta una estructura diferente, organizada en 38 clases. Una obra pionera en la terminología, que a su vez cae en esta clasificación, es el diccionario multilingüe *The Machine Tool* [Wuster, 1968].

Los pasos para encontrar una palabra en los tesauros son: identificar una idea general del concepto, seleccionar una palabra clave como punto de partida. Posteriormente, se toman un subconjunto de palabras que sean más significativas para hacer la búsqueda. Este proceso es similar a cuando se hacen búsquedas de un concepto en un motor de recuperación de información, de tal forma que se seleccionan las palabras que se creen están más cerca al concepto y

pueden obtener el resultado deseado.

Es importante mencionar que dicho proceso tiene algunas desventajas, es posible que los usuarios no encuentren las palabras adecuadas que concuerden con las palabras clave del tesauro. Otra desventaja, es la forma de clasificación que puede tornarse sumamente compleja, siendo así, que para subsanar este problema los tesauros incluyen un índice ordenado alfabéticamente, que según algunos autores, resulta ser el mejor punto de partida en las búsquedas.

3.2.1.2. Diccionarios de sinónimos y antónimos

La mayoría de los lexicógrafos concuerdan que los diccionarios de sinónimos son un tipo de diccionarios onomasiológicos. La meta consiste en aumentar el vocabulario y encontrar nuevas relaciones de asociación. Así, se encuentran otras palabras que pueden ser intercambiadas con las originales, conservando el mismo significado. De esta forma, se encuentran palabras que se han olvidado o se desconocen. Para llevar a cabo esta búsqueda, se deben identificar palabras clave con significado similar al de la palabra destino. Este tipo de diccionarios presentan listas de palabras relacionadas, en la mayoría de los casos, las entradas están en orden alfabético y las listas internas se pueden organizar también de manera alfabética aunque no necesariamente. En algunas ocasiones se presentan antónimos, sin embargo también existen diccionarios de antónimos.

Una característica que comparten con los tesauros, es que la mayoría de los diccionarios de sinónimos presentan sus palabras relacionadas pero no proporcionan el significado de ellas. Solamente algunos de ellos establecen el significado de los sinónimos presentados y proporcionan las posibles diferencias en su uso. Sin embargo, lo que hace diferente a un diccionario de sinónimos comparado con un tesauro, es que el primero opera sobre palabras, en vez de conceptos, como lo hacen los tesauros [Alvar Esquerra, 1984].

Para el idioma inglés, es importante mencionar el *Webster's New Dictionary of Synonyms* [1984], y para español, el Diccionario de Sinónimos [Barcia, 2000]. Desafortunadamente, parece que los diccionarios de sinónimos/antónimos no son las herramientas más adecuadas para encontrar el concepto objetivo.

3.2.1.3. Diccionarios visuales

Son también llamados diccionarios pictóricos, sus características particulares les permiten ser mejores que otros libros de palabras. Al igual que en los diccionarios conceptuales, la realidad se presenta ordenada por conceptos, pero en este caso los conceptos se muestran con dibujos que ilustran las partes o especies del concepto. Posteriormente, una palabra muestra el nombre de la parte o la especie. No se presenta una definición explícita ya que visualmente se puede observar la relación entre el nombre y el objeto. El diccionario presenta en sus páginas el vocabulario de temas completos, agrupados de acuerdo a una clasificación específica. Generalmente contienen índices alfabéticos que facilitan las búsquedas partiendo de una palabra al objeto. Las limitantes de este tipo de diccionarios es que solo permiten representar objetos físicos y sus partes o especies. Por lo general solo se representan sustantivos, aunque algunas veces también se ilustran verbos que muestran acciones y algunos adjetivos. La información presentada, así como la temática de las imágenes están muy relacionadas con la imaginación de los dibujantes. Permitiendo que el diccionario sea visto a manera de enciclopedia, por ejemplo, es posible encontrar toda una variedad de 50 tipos de aves [2].

El diccionario de Oxford-Duden [Alvar Ezquerra *et al.*, 1995] es uno de los más conocidos. Existen versiones bilingües basadas en el *Bildwörterbuch* (Diccionario de imágenes) de origen alemán, cuya primera edición fue realizada en 1937. Se encuentra estructurado semánticamente, agrupándose en conjuntos de 11 a 15 temas dependiendo de la versión, contiene alrededor de 400 páginas, abarcando aproximadamente 30,000 palabras.

Aunque los diccionarios visuales tienen un planteamiento onomasiológico, principalmente porque permiten encontrar una palabra objetivo a través de la observación de una imagen del concepto, algunos autores difieren en el propósito de ellos. Shcherba [1995] establece que este tipo de búsquedas son invaluables cuando se trata de encontrar una palabra en un idioma extranjero, por otra parte Hill [1985] establece que estos diccionarios son de utilidad para profesores y escritores, más que para estudiar un idioma extranjero.

[2] http://www.ikonet.com/es/diccionariovisual/reino-animal/aves/

3.2.1.4. Diccionarios inversos

Permiten a los usuarios la búsqueda desde una palabra pista en vez de un índice o un árbol conceptual. Para encontrar una palabra destino en el diccionario, los usuarios piensan en un concepto y en una palabra pista que refiera a él, para posteriormente ir a la parte principal del diccionario, el "diccionario inverso". La macro estructura es alfabética, lo cual permite a los usuarios ir desde la palabra pista al concepto sin un índice. Cada palabra pista tiene una lista reducida de palabras relacionadas seguido de una breve definición de cada concepto. Sin embargo, estos recursos tienen varias dificultades. Primero, puede ser que no exista una palabra pista adecuada, y segundo, puede suceder que la palabra pista sí exista en un diccionario, pero no está ligada al objetivo buscado.

Algunos ejemplos de diccionarios inversos son: *Bernstein's Reverse Dictionary* [1975], *Reader's Digest Reverse Dictionary* [1989] y *The Oxford Reverse Dictionary* [2002].

En el diccionario Bernstein se tienen 13,390 entradas que son accesibles por medio de 8,000 palabras claves. Estos datos sugieren un acceso muy limitado a cada entrada, ya que son aproximadamente 2 palabras por cada pista. Restringiendo las posibilidades de identificar los conceptos. Tomando en cuenta este tipo de posibles problemáticas, el diccionario Digest alienta a realizar búsquedas con diferentes claves, maximizando las probabilidades de obtener el concepto deseado.

3.2.2. Diccionarios onomasiológicos electrónicos

Un importante representante de los tesauros electrónicos es WordNet[3] [Miller, 1990], quizá no puede ser clasificado como un diccionario onomasiológico electrónico *per se*, pero debido a su estructura y organización se puede contemplar como una aproximación a este tipo de diccionarios. Desarrollado en la Universidad de Princeton, su organización contempla diferentes relaciones semánticas: hiperonimia, hiponimia, sinonimia, holonimia ("forma parte de") y meronimia ("tiene un"). WordNet está estructurada como una gran base de datos léxica de inglés. Todas las palabras pertenecientes a Wordnet tienen asociada una categoría gramatical y se

[3]https://wordnet.princeton.edu/

encuentran agrupadas en conjuntos de sinónimos cognitivos, denominados *synsets*. Cada uno de estos elementos expresa un concepto distinto y se encuentran interconectados por medio de relaciones conceptual-semánticas y léxicas.

Todo el escenario de las búsquedas onomasiológicas cambió con la universalización del Internet y las tecnologías del lenguaje, permitiendo la construcción de recursos en línea, en conjunto con los grandes volúmenes de información. Las limitaciones de los diccionarios impresos, hoy día se encuentran superadas gracias al desarrollo de la tecnología, en particular de la lexicografía computacional. Los avances en esta área son dignos de conferencias internacionales, como es el caso del *ELEX (Electronic Lexicography)*, donde se presentan los aportes a la lexicografía en sinergia con la perspectiva evolutiva de los alcances tecnológicos en turno. En las últimas dos décadas, diversos diccionarios en línea han sido diseñados para búsquedas en lenguaje natural. Los usuarios introducen sus propias definiciones en lenguaje natural y los motores buscan las palabras que concuerdan con la definición. Uno de los primeros diccionarios en línea que permite este tipo de búsquedas es el creado por Dutoit y Nugues [2002]. El recurso toma en cuenta las diferencias entre usuarios regulares y diccionarios formales cuando se describe un término. En el diccionario se usa una base de datos jerárquicamente organizada, de tal forma que los hiperónimos e hipónimos están automáticamente identificados. Una de las desventajas de este sistema es que la sinonimia no está considerada. Bilac *et al.* [2004] diseñaron un diccionario para el idioma Japonés en el que los usuarios pueden introducir sus definiciones. Tiene un algoritmo que calcula la similitud entre conceptos comparando las palabras. Tal medida decrece cuando la definición contiene palabras que no son exactamente aquellas que están presentes en la base de datos, siendo esto uno de los principales problemas de esta aplicación web. El-Kahlout y Oflazer [2004] construyeron un recurso similar para el idioma Turco. Tomaron en cuenta algunas relaciones de sinonimia entre palabras, así como la similitud de las definiciones por medio de un contador de palabras similares en el mismo orden y un subconjunto de tales palabras. Los resultados son en su mayoría positivos: 66 % de las veces el término es encontrado entre los primeros 50 candidatos. Cuando se usan las definiciones de otros diccionarios como entrada, la puntuación alcanza el 92 % dentro de los primeros 50. Sin embargo, esta implementación no

toma en cuenta el uso de coloquialismos; el número de términos candidatos, 50, es muy alto, y no toma en cuenta la posición promedio de los conceptos objetivo en la lista.

Uno de los principales trabajos en Español es el desarrollado por Sierra [2000a]. DEBO es un diccionario onomasiológico que trabaja con consultas de usuarios en lenguaje natural y un motor de búsqueda. El algoritmo consiste en encontrar coincidencias entre las palabras de la consulta que establece el usuario y las palabras que tiene en su base de conocimientos. Una vez obtenida la consulta, se remueven las palabras vacías (*stopwords*) para seleccionar únicamente aquellas que sean importantes para la definición y se lematizan. Este diccionario trabaja con paradigmas, es decir, grupos de palabras con significados similares. Posteriormente, se detectan los paradigmas a los que pertenecen las palabras y localiza aquellos términos que estén relacionados con los mismos paradigmas presentes en la consulta del usuario, devolviéndolos como resultado final del algoritmo. La lista obtenida se ordena de acuerdo a la cantidad de paradigmas que se tengan en común, cuando hay un empate en el número de ellos, se realiza un ordenamiento alfabético.

El algoritmo fue mejorado por Hernández [2012], quien también optimizó la estructura de la base de datos.

Para la evaluación, se usaron definiciones de usuarios regulares y se calculó el promedio de la palabra objetivo. El algoritmo de Hernández mejoró por 15 % los resultados iniciales de Sierra. Comparado con otros trabajos, como el de El-Kahlout y Oflazer [2004], este motor de búsqueda mejoró los resultados por 5 %.

3.2.2.1. Diccionarios inversos basados en redes neuronales

Los diccionarios onomasiológicos han sido abordados también desde la perspectiva de las redes neuronales, haciendo uso de modelos de vectores de palabras o *word embeddings*. Los detalles de estos modelos vectoriales se establecen en el capítulo 4.

Hill *et al.* [2016] presentan la creación de un diccionario inverso modelando vectores de palabras. Para el entrenamiento usan las definiciones de diccionarios y enciclopedias; el modelo está basado en dos aproximaciones: (1) una red neuronal recurrente (*Recurrent Neural Networks o RNN*) con una LSTM (*long-short-term-memory*) que codifica naturalmente el or-

den de las palabras, y (2) un modelo vectorial simple (propagación hacia adelante) de bolsa de palabras (*bag-of-words o BOW*). Los resultados de esta propuesta fueron comparados con aplicaciones comerciales de diccionarios inversos, como *OneLook* y *Dictionary.com*, mostrando que sus resultados son competitivos y en algunas ocasiones incluso mejores.

Zheng *et al.* [2020] proponen un modelo de diccionario inverso multicanal, el cual se inspira en el proceso de inferencia al describir palabras, que realizan los humanos. Los autores emplean múltiples predictores para identificar diferentes características de las palabras objetivo desde la consulta de entrada. En total se usan cuatro canales, los canales internos corresponden a las características de las palabras que incluyen las etiquetas POS (*Part-Of-Speech*) y los morfemas; los canales externos incluyen características de las palabras objetivo, que corresponden a la categoría de la palabra y al semema, este último se define como la unidad semántica mínima [Bloomfield, 1926]. Para su implementación se usó un marco de trabajo basado en *LSTM* bidireccional [Hochreiter y Schmidhuber, 1997] con atención [Bahdanau *et al.*, 2014], añadiéndole cuatro predictores de características. WantWords permite consultas en línea[4] tanto para Chino como Inglés, sin embargo, esta implementación sustituye el codificador de la oración y en vez de *BiLSTM* se usó un modelo basado en *BERT* como única diferencia respecto al trabajo original.

3.2.2.2. Diccionarios inversos basados en grafos

La teoría de grafos también ha servido para la creación de diccionarios inversos. Se utilizan representaciones semánticas en este tipo de objeto matemático y se aprovechan las relaciones establecidas para realizar búsquedas, aplicando algoritmos propios del área. Detalles de estos objetos matemáticos serán abordados en el capítulo 4.

Un modelo muy intuitivo para resolver este problema es el uso de redes. La solución simple a la necesidad de tener relaciones sintagmáticas-paradigmáticas entre las palabras puede ser el uso de redes de colocación [Ferrer i Cancho y Solé, 2001]. Los autores usaron el corpus de *British National Corpus (BNC)* para construir dos grafos: G1 y G2. Primero, un grafo de co-ocurrencias G1 en el cual las palabras están ligadas si ocurren en al menos una oración con un

[4] `https://wantwords.thunlp.org/`

salto máximo de 3 *tokens*. Posteriormente, un grafo de colocación G2 es extraído, en el cual únicamente se conservan los lados de aquellas ligas de G1 en las que los vértices finales co-ocurren con mayor frecuencia de lo esperado. Widdows y Dorow [2002] sugirieron la posibilidad de limitar el corpus con etiquetas PoS (*part of speech*). El grafo debe estar anotado de acuerdo al criterio seguido para construir la red. Zock *et al.* [2010] usó las etiquetas AKO(*a kind of*), ISA(subtipo), TIORA(*Typically involved object*, relación o actor). Dado que el grafo también involucra ligas sintagmáticas, más etiquetas deben ser introducidas para describir las relaciones sintácticas.

Thorat y Choudhari [2016] proponen un diccionario basado en un grafo cuyos nodos son obtenidos de las información semántica extraída de definiciones. El algoritmo usa una medida de similitud basada en distancia para evaluar la similitud entre una frase y la palabra objetivo. Los resultados que presenta esta propuesta, establecen que son comparables con *OneLook*.

Las Normas de Asociación de Palabras presentadas en el capítulo 2 pueden ser tratadas como un grafo, estableciendo la arista que une dos vértices como la relación entre un estímulo con su correspondiente respuesta. Las NAP son un tipo especial grafo de conocimiento (*Knowledge graph*) y están disponibles en muchos idiomas. Con base en estos grafos, la presente tesis propone un modelo para realizar búsquedas. Los resultados obtenidos han sido publicados en Reyes-Magaña *et al.* [2019a] y en Reyes-Magaña *et al.* [2019b], en el primero se utilizó el corpus de Normas de Asociación de Palabra para el Español de México, y en el segundo se trabajó con los corpus de *Edinburgh Associative Thesaurus (EAT)* y la colección de la *University of South Florida*. Los dos trabajos utilizan algoritmos de centralidad basados en grafos, mostrando resultados que superan *OneLook*, así como otros algoritmos que se consideraron adecuados como parámetro de comparación. La descripción de los métodos se hace más adelante en los capítulos 5 y 6.

3.2.2.3. Diccionario inverso *OneLook*

Para el idioma Inglés, existe un diccionario inverso en línea, *OneLook Reverse Dictionary*[5], que permite realizar consultas en lenguaje natural y también trabaja con expresiones regulares. Este diccionario se utiliza generalmente como punto de comparación para la evaluación de resultados en tareas de diccionarios inversos. Es de código cerrado, y la arquitectura interna es desconocida. Sin embargo, posee una API que permite realizar consultas a través de código. Es importante mencionar que este diccionario posee una versión en español, pero sus resultados son poco favorables.

3.3. Modelos tradicionales de recuperación de información

Un diccionario inverso puede ser considerado como una tarea de **recuperación de información** o *Retrieval Information*, la cual es un área importante de PLN.

La recuperación de información consiste en encontrar material (usualmente documentos) de naturaleza no estructurada (usualmente texto) que satisface información necesitada de una colección de gran tamaño (usualmente almacenada en computadoras) [Manning *et al.*, 2009].

Al analizar la definición formal de un sistema de recuperación de información, se puede contrastar con la problemática de búsqueda léxica, como se describe a continuación:

- **Encontrar material (usualmente documentos).** La idea principal es poder llegar a las palabras partiendo de la descripción o un tipo de definición dada por una persona.
- **De naturaleza no estructurada (usualmente texto).** Las definiciones proporcionadas por las personas, son además de estar en formato texto, de una naturaleza no científica, sin tener un rigor formal en su descripción.
- **Que satisface información necesitada.** Este tipo de búsquedas salen de la necesitad de palabras, que de algún modo, se escapan de la mente, como el problema de la punta de la lengua. Son de utilidad para los escritores o cualquier persona que se olvide de la

[5] https://www.onelook.com/reverse-dictionary.shtml

palabra, y a pesar de tener el término en la mente, en su momento solo se presenta la idea que lo describe.

- **De una colección de gran tamaño (usualmente almacenada en computadoras).** Las colecciones de datos que son usadas para resolver el acceso léxico, son corpus digitales, y dentro de esta basta colección se trata de encontrar aquella palabra que satisface la consulta de una persona. Esto se puede hacer de diferentes maneras. En el caso particular de esta tesis se usaron algoritmos basados en grafos.

Con base en lo anterior, podemos afirmar que la creación del modelo de búsqueda léxica, es parte del procesamiento de lenguaje natural en lo que respecta al área de *information retrieval.*

Dentro de los modelos clásicos de recuperación de información se encuentra el modelo booleano. Hoy día este modelo es ampliamente reconocido y entendido [Lancaster, 1979]. Se basa en la intersección o unión de listas para implementar la conjunción (A **Y** B) o la disyunción (A **O** B) respectivamente. En este modelo se genera una consulta unida por operadores booleanos, por ejemplo, suponiendo que tenemos la siguiente consulta:

animal que ruge y es el rey de la selva

En términos de un sistema recuperación de información tipo booleano, generaría una consulta:

animal ***Y*** *que* ***Y*** *ruge* ***Y*** *y* ***Y*** *es* ***Y*** *el* ***Y*** *rey* ***Y*** *de* ***Y*** *la* ***Y*** *selva*

Los operadores que unen las palabras de la consulta pueden ser tipo conjunción (Y), disyunción (O) o negación (NO).

Con la sentencia booleana construida se procede a buscar en la colección de documentos, presentando aquellos que cumplan con la lógica indicada en la consulta. En algunos casos es mejor seleccionar aquellas palabras que sean las más representativas de la consulta, es decir, las que no son funcionales y posteriormente construir la consulta.

Uno de los algoritmos más exitosos de recuperación de información es *Okapi BM25*, el cual está basado en un modelo probabilístico, fue desarrollado en los setentas por Robertson

y Sparck Jones [1976]. La implementación de Robertson y Zaragoza [2009] está basada en el modelo de bolsa de palabras. Dada una consulta, se ordena una lista de documentos con base en su relevancia para tal consulta.

Capítulo 4

Técnicas de Procesamiento de Lenguaje Natural

Con la finalidad de realizar procesamiento de textos de manera automática en sistemas computacionales, es necesaria una serie de representaciones de los textos. En este capítulo se presenta una diversidad de formas de representación de textos que han demostrado ser valiosas en la captura del sentido de las palabras y de sus diversas interrelaciones. Con estas se establecen elementos entendibles para los sistemas computacionales y de esta forma se permite un adecuado procesamiento automático.

Se hará un mayor énfasis en el desarrollo de algunos elementos teóricos de la teoría de grafos, ya que la metodología abordada en el modelo de búsqueda léxica utiliza en su mayoría conceptos de esta teoría.

4.1. Fundamentos de PLN

El desarrollo actual de Internet y, sobre todo, el uso de redes sociales ha traído un volumen de información vertiginoso. En el caso de la información basada en texto, le agrega un componente de complejidad dada su característica no estructurada. Es así que surge la necesidad del procesamiento automático de la información, en formato texto, que satisfaga necesidades, como

por ejemplo: velocidad de procesamiento, confiabilidad y precisión en los resultados para la tarea en cuestión, por mencionar algunos. Esto deriva en una gran cantidad de aplicaciones para la vida actual que, con el paso del tiempo, y gracias al desarrollo de nuevos aportes de la Inteligencia Artificial, se han ido modificando y perfeccionando en sus técnicas para proporcionar mejores resultados.

De acuerdo con Liddy [2001], el procesamiento de Lenguaje natural se define como: una serie de técnicas computacionales teóricamente motivadas para analizar y representar textos en uno o más niveles de análisis lingüísticos, con el propósito de lograr un procesamiento de lenguaje similar al que harían los humanos para un rango de tareas y aplicaciones.

Existe una basta diversidad de representaciones computacionales para la información en formato texto, a continuación se presentan las más importantes y sobre todo, las que se usaron en el desarrollo de esta investigación y los temas relacionados a la misma.

4.1.1. N-gramas

Los n-gramas de textos son usados extensamente en minería de textos, la cual se define como el proceso de analizar texto para extraer información que es útil para propósitos particulares [Witten, 2004].

Son un conjunto de elementos consecutivos dentro de un rango de continuidad determinada; es decir, algunas veces la continuidad permite saltos entre los elementos, aunque normalmente los n-gramas se toman sin salto alguno.

Los n-gramas y todas sus variantes son un conjunto de elementos consecutivos en una misma oración. El valor de n cambia desde 1 hasta n, dando origen a unigramas (elementos por separado), bigramas, trigramas, etc.

De manera general los n-gramas se pueden construir de una variedad de elementos, no solamente considerando palabras, que podría considerarse lo más común, también pueden ser de caracteres, palabras funcionales (*stopwords*), signos de puntuación, etiquetas PoS (*Part of the Speech*), o incluso de símbolos de puntuación. Todos ellos parten de la misma idea descrita anteriormente, el único elemento cambiante es el tipo de unidad tomada para su construcción.

Los n-gramas son usados para una variedad de tareas diferentes. *Google*[1] y *Microsoft*[2] han desarrollado modelos de n-gramas que pueden ser usados en una gran variedad de tareas, tales como corrección ortográfica, separación de palabras y resúmenes de texto. Ganesan *et al.* [2012] presentan un trabajo que usa modelos de n-gramas para resúmenes de texto.

Otro uso de los n-gramas es el desarrollo de características para modelos de aprendizaje máquina supervisado, tales como SVM (*Support Vector Machine*) [Scholkopf y Smola, 2001], modelos MaxEnt (*Maximun Entropy*) [Ratnaparkhi, 1997] y Naive Bayes [Marmanis y Babenko, 2009], entre otros.

4.1.1.1. Palabras

Como su nombre lo indica, los n-gramas de palabras se construyen para establecer conjuntos de palabras que co-ocurren en una misma oración. La tabla 4-1 presenta los bigramas de la oración "Hoy es un excelente día" cuando n=2.

Hoy es
es un
un excelente
excelente día

Tabla 4-1: Bigramas de palabras

La cantidad de palabras tomadas para construir los n-gramas varía dependiendo de la tarea de procesamiento de lenguaje natural, en alguna ocasiones se realizan varias pruebas para determinar el tamaño conveniente que obtenga los mejores resultados.

Una aplicación de su uso es la atribución de autoría que permite identificar documentos de autores anónimos; por ejemplo, Wright [2017] utiliza n-gramas de palabras para identificar la autoría en *emails* de usuarios anónimos. En el trabajo presentado por Doddington [2002], se evalúa la calidad de una traducción identificando la cantidad de n-gramas que comparte con respecto a la traducción de referencia, es decir, que mientras mayor sea la cantidad de n-gramas,

[1] `https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html`
[2] `https://www.microsoft.com/en-us/research/project/web-n-gram-services/`

entonces se establece que la calidad es mejor.

4.1.1.2. Caracteres

Los n-gramas de caracteres se encuentran en documentos de textos por la representación del documento como una secuencia de caracteres. Los n-gramas se extraen de esta secuencia y con ello se entrena un modelo [Layton, 2015].

En la frase "Procesamiento de lenguaje natural", la tabla 4-2 muestra n-gramas de caracteres dependiendo de la cantidad de caracteres tomados para su construcción.

n	n-gramas
2	Pr,ro,oc,ce,es,sa,....
3	Pro,roc,oce,ces,esa,sam,...
4	Proc,roce,oces,cesa,esam,...

Tabla 4-2: n-gramas de caracteres, para n=2,3,4

Sapkota *et al.* [2015] establecen que este tipo de n-gramas han sido identificados como la característica más exitosa en la atribución de autoría. Permiten capturar aspectos lingüísticos correspondientes a elementos como la morfosintáxis, contenido y estilo.

Para tareas de Procesamiento de Lenguaje Natural, donde es probable que muchas palabras estén mal escritas, los n-gramas de caracteres son especialmente poderosos [Sanchez-Perez *et al.*, 2017].

4.1.1.3. Etiquetas PoS (*Part of the Speech*)

Una de las partes más fundamentales de la lingüística es el etiquetado PoS (Parte de la oración), que representa una forma básica del análisis sintáctico, teniendo una gran cantidad de aplicaciones en el procesamiento del lenguaje natural [Gimpel *et al.*, 2010].

Existen herramientas computacionales de código abierto, como *Freeling* [Padró y Stanilovsky, 2012] que proporciona un variedad de analizadores para diversas lenguas. Identifica categorías gramaticales de palabras estableciéndolas como artículos, pronombres, sustantivos,

pronombres, adjetivos, adverbios, verbos, etc. Para ello utiliza etiquetas con una sintaxis específica basada en lo propuesto por EAGLES *(Expert Advisory Groups on Language Engineering Standards)* [Leech y Wilson, 1999].

EAGLES tiene la intención de poder codificar todas los rasgos morfológicos de la mayoría de los lenguajes europeos. Las etiquetas son de longitud variable, donde cada caracter corresponde a un rasgo morfológico. El primer caracter siempre se refiere a la categoría. La categoría determina la longitud de la etiqueta y la interpretación de cada caracter en la etiqueta. Por ejemplo, los elementos presentes en una etiqueta de la categoría "sustantivo" para el español, está representada por la tabla 4-3, que muestra cada uno de los rasgos morfológicos correspondientes a esta categoría gramatical. Siguiendo esta sintaxis posicional, cada categoría gramatical tiene sus propios elementos que la definen. Especificaciones más detalladas de la composición de las etiquetas puede ser consultada en el analizador morfológico del español[3].

Posición	Atributo	Valores
0	categoria	N:sustantivo
1	tipo	C:común;P:propio
2	género	F:femenino; M:masculino; C:común
3	número	S:singular; P:plural; N:invariable
4	neclass	S:persona; G:sitio;O:organización;V:otra
5	nesubclass	sin uso
6	grado	V:evaluativo

Tabla 4-3: Etiqueta *EAGLES* para **sustantivo**

Las etiquetas generadas para la frase ' "Procesamiento de lenguaje natural" son: NCMS000 SP NCMS000 AQ0CS00. Por lo que en un texto completo, es posible generar un documento que contenga las etiquetas *POS* que representen cada una de la palabras del documento original y de esta forma construir n-gramas basados en categorías gramaticales.

El uso de este tipo de n-gramas es de gran utilidad, ya que al captar información sintáctica permite identificar la intención de los usuarios en tuits; un ejemplo de ello es el trabajo

[3] `https://www.cs.upc.edu/~nlp/tools/parole-sp.html`

presentado por Gómez-Adorno *et al.* [2018], que detecta agresividad en Twitter™.

4.1.1.4. Skip-grams

Los skip-grams son una técnica usada en el campo de procesamiento de discurso, mediante el cual se forman n-gramas (bigramas, trigramas, etc.), pero que, además de establecer secuencias de palabras, permite que los elementos sean "saltados". Aunque inicialmente se aplicaba a los fonemas en el habla humana, la misma técnica se puede aplicar a las palabras [Guthrie *et al.*, 2006].

De realizar esta técnica es posible encontrar conjuntos significativos de palabras, ayudando a la identificación de patrones en los textos que mediante el uso de n-gramas de palabras adyacentes sería difícil.

En la siguiente fórmula se establece la definición matemática del conjunto que representa un *skipgram*.

$$\{w_{i_1}, w_{i_2}, ..., w_{i_n} | \sum_{j=1}^{n} i_j - i_{j-1} < k\} \text{(4-1)}$$

Los skip-grams establecidos para una distancia de salto k permiten un total de k o menos saltos para construir el n-grama. Como tal, un "4-skip-n-grama" resulta en la inclusión de 4, 3, 2, 1 y 0 saltos, estos últimos representando los n-gramas típicos formados de palabras adyacentes.

Se presenta un ejemplo mostrando *2-skip-bigramas* y *2-skip-trigramas*, comparado con bigramas y trigramas de palabras adyacentes para la oración:

El procesamiento de lenguaje natural

- **Bigramas** = { El procesamiento, procesamiento de, de lenguaje, lenguaje natural }.
- **2-skip-bigramas** = { El procesamiento, El de, El lenguaje, procesamiento de, procesamiento lenguaje, procesamiento natural, de lenguaje, de natural, lenguaje natural}.
- **trigramas** = { El procesamiento de, procesamiento de lenguaje, de lenguaje natural}.

- **2-skip-trigramas** = { El procesamiento de, El procesamiento lenguaje, El procesamiento natural, El de lenguaje, El de natural, El lenguaje natural,procesamiento de lenguaje, procesamiento de natural, procesamiento lenguaje natural, de lenguaje natural}.

Sidorov *et al.* [2014], presentan un trabajo que usa skip-grams para identificar lenguaje nativo, los resultados sugieren que los skip-grams son muy útiles en este tipo de tarea. También pueden ser usados como sustitución de dependencias en escenarios donde ejecutar un analizador completo no puede ser factible, como por ejemplo un procesador de datos en tiempo real.

4.1.1.5. Otros tipos de ngramas

Los algoritmos de aprendizaje máquina permiten obtener patrones de acuerdo a los datos que son introducidos, de esta forma mientras más patrones puedan ser localizados en el texto, es mucho mejor para la solución de la tarea a resolver. De acuerdo con Argota Vega *et al.* [2019], entre los n-gramas que también son usados se encuentran:

- **Palabras funcionales.** Las palabras funcionales son aquellas como los artículos, preposiciones, pronombres, etc. que no aportan un significado por sí mismas, pero que en conjunto con otro tipo de palabras de contenido, permiten el entendimiento de la oración completa.

- **Símbolos de puntuación.** La construcción de n-gramas con estos símbolos permite identificar las intenciones de los usuarios. En casos como detección de agresividad y odio, se ha identificado que aquellos textos que representan mayor énfasis en la agresividad del texto, presentan una mayor de símbolos de puntuación, un ejemplo de ellos son los símbolos de exclamación como se puede observar en el siguiente tuit:

> El ministro Alejandro Finocchiaro @alefinocchiaro nos prometía Finlandia en Educación hace poco con @pvesterbacka colaborando! Hoy con el AJUSTE lejos de parecernos a Finlandia estamos más cerca de parecernos a cualquier país subsahariano **....!!!** Lástima no??

- **Palabras específicas.** En el trabajo de Gómez-Adorno *et al.* [2018] se usan n-gramas de palabras agresivas para detectar sentimientos en redes sociales. Si el texto contiene palabras que pertenezcan al diccionario de palabras agresivas se establece el *n-grama* con solamente dichas palabras.

4.2. Vectores de palabras

La representación semántica de palabras en un espacio vectorial es un área de investigación muy activa en las últimas décadas. Modelos computacionales como la descomposición de valores singulares (SVD) y el análisis semántico latente (LSA) son capaces de modelar representaciones continuas de palabras (*word embeddings*) a partir de matrices término-documento. Ambos métodos pueden reducir un conjunto de datos de N dimensiones utilizando solo las dimensiones más importantes. Recientemente, Mikolov *et al.* [2013] introdujeron *word2vec*, inspirado en la hipótesis distribucional que establece que las palabras en contextos similares tienden a tener significados similares [Sahlgren, 2008]. Dicho método utiliza una red neuronal para aprender representaciones vectoriales de palabras al predecir otras palabras en su contexto. La representación vectorial de la palabra obtenida mediante *word2vec* tiene la asombrosa capacidad de preservar las regularidades lineales entre palabras. Para construir un modelo de espacio vectorial adecuado y confiable, capaz de capturar la similitud semántica y las regularidades lineales de las palabras, se necesitan grandes volúmenes de texto. Los vectores pre-entrenados generalmente están disponibles en línea. En la tabla 4-4 se presentan los recursos *FastText*[4] [5], *Glove* [6] y *Word2vec*[7] para el idioma español, con sus principales características.

Es posible realizar el entrenamiento de estos vectores con cualquier tipo de corpus basado en texto, mientras mayor sea el tamaño del corpus, la representatividad de cada vector será mejor.

Para las diferentes tareas de procesamiento de lenguaje natural, es muy útil el uso de vectores

[4] `http://dcc.uchile.cl/~jperez/word-embeddings/fasttext-sbwc.vec.gz`
[5] `https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.es.vec`
[6] `http://dcc.uchile.cl/~jperez/word-embeddings/glove-sbwc.i25.vec.gz`
[7] `http://cs.famaf.unc.edu.ar/~ccardellino/SBWCE/SBW-vectors-300-min5.txt.bz2`

Nombre	Corpus	Algoritmo
FastText [Bojanowski *et al.*, 2017]	*Spanish Billion Word Corpus* (1.5 billones)	Skipgram
Glove [Pennington *et al.*, 2014]	*Spanish Billion Word Corpus* (1.5 billones)	GloVe
FastText [Bojanowski *et al.*, 2017]	Wikipedia en español	Skipgram
Word2Vec [Mikolov *et al.*, 2013]	*Spanish Billion Word Corpus* (1.5 billones)	Skipgram

Tabla 4-4: Vectores pre-entrenados en español

de palabras, algunos trabajos que utilizan este tipo de tecnología se presentan a continuación:

- Zhang *et al.* [2015], estudian el desarrollo emergente del comercio electrónico, por medio de un análisis en los comentarios que proporcionan los clientes al realizar sus compras en línea. Se hace una investigación basada en análisis de sentimientos, es decir se enfoca en la clasificación de comentarios como positivos o negativos de acuerdo con la polaridad del sentimiento. El aprendizaje máquina se ha vuelto de los principales métodos para realizar esta tarea por lo que este trabajo analiza las características semánticas por medio de vectores de *word2vec* y máquinas de soporte vectorial.

- El objetivo de la construcción de relaciones semánticas jerárquicas es la identificación de conceptos ligados por relaciones de *hiperonimia - hiponimia* ("es un"). Por ejemplo, el concepto médico es el hiperónimo de un anestesiólogo, y de manera inversa un anestesiólogo es un hipónimo de médico. Un reto mayor para esta tarea es el descubrimiento automático de tales relaciones. El trabajo de Fu *et al.* [2014], propone un método novedoso y efectivo para la construcción de jerarquías semánticas basadas en vectores de palabras, que pueden ser usados para medir la relación semántica entre palabras. Se identificó cuándo un candidato de un par de palabras tiene una relación de *hiperonimia-hiponimia* por medio de las proyecciones semánticas basadas en vectores.

Los *WordEmbeddings* se usaron para la validación del modelo de búsqueda léxica desarrollado en esta tesis. Más adelante se describe el algoritmo a detalle.

4.3. Grafos

El primer artículo escrito sobre teoría de grafos fue en 1736 por Leonhard Euler, que estudió el problema de los siete puentes de Könisberg (hoy Kaliningrado, Rusia) de cómo visitar cada área de la ciudad cruzando cada puente solamente una vez [Zuhair *et al.*, 2017]. Könisberg era una ciudad en Prusia con el río Pregel que lo atravesaba, creando dos islas. La ciudad y las islas estaban conectadas por siete puentes como se aprecia en la figura 4.3.

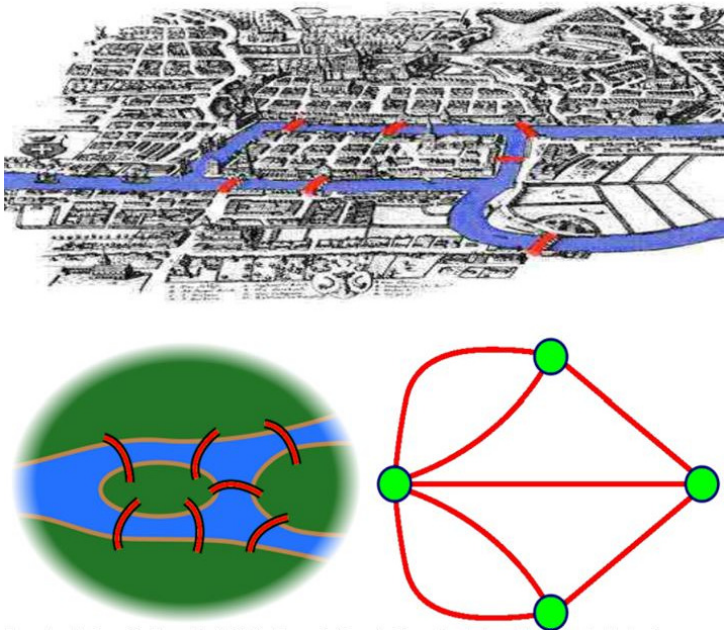

Figura 4-1: Puentes de Könisberg [Calero Medina, 2012].

Euler formuló una abstracción del problema, eliminando hechos innecesarios y enfocándose en las área de tierra y los puentes que las conectaban. Encontró que empezar la ruta desde cualquier área de tierra es irrelevante y lo único importante es el orden de recorrido. Encontró que el problema no tenía solución y se requería un octavo puente para ser construido. Su trabajo dio origen a la teoría de grafos.

El uso de teoría de grafos en el estudio de relaciones fue comenzado por un grupo de matemáticos y psicólogos la década de los 40's. La teoría de grafos permite a los investigadores probar teoremas y deducir afirmaciones. Un **grafo** es un objeto matemático que describe re-

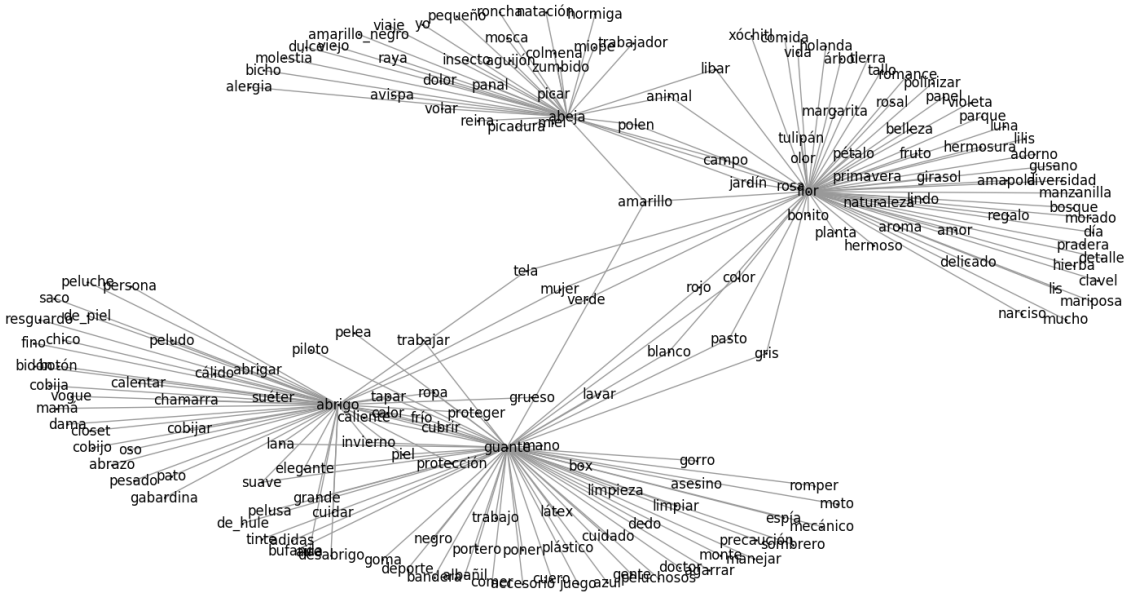

Figura 4-2: Grafo con los estímulos flor, abeja, abrigo y guante con sus correspondientes asociados.

laciones entre elementos. Es una estructura para modelar información. Consiste de nodos que representan objetos y lados que relacionan un nodo con otro.

La teoría de grafos ha tenido un gran impacto en computación, ya que permite representar diversas relaciones del mundo real: amigos en una red social, páginas web, células en una red neuronal, etc. Muchos problemas prácticos han sido modelados como grafos. Por ejemplo, han sido usados para modelar situaciones en física, biología, psicología, ciencias computacionales y en particular tareas de procesamiento de lenguaje natural, el cual es el tema principal para el desarrollo de este trabajo.

La figura 4.3 muestra un grafo construido con algunas palabras del NAP, que representan *estímulos* y correspondientes *respuestas*.

Zuhair *et al.* [2017] establecen que las clases de problemas relacionados al estudio de grafos se pueden agrupar en:

1. *Existencia:* problemas que tratan de determinar si un camino, un vértice o un conjunto

existen.

2. *Construcción:* dado un conjunto de caminos y vértices, y dada una restricción, ¿cómo construir un grafo?

3. *Enumeración:* problemas que tratan de determinar cuantos vértices y lados existen, dado un conjunto de restricciones.

4. *Optimización:* problemas que tratan de encontrar el camino más corto entre dos nodos.

El modelo de búsqueda léxica propuesto en esta tesis es una combinación de un problema de *existencia* combinado con *optimización* y *enumeración*, ya que permite la identificación de nodos, significantes, que respondan a una definición o significado. Para ello, las palabras de la definición se configuran como un camino que pueda dirigir la búsqueda hasta el mejor nodo relacionado.

De manera formal un grafo G con un conjunto X de vértices y un conjunto de lados E, se escribe como $G=(X,E)$. La figura 4.3 presenta un grafo con 4 nodos y 4 lados, y cae en la clasificación de un grafo no dirigido, ya que no hay un sentido de inicio y final en cada lado del grafo.

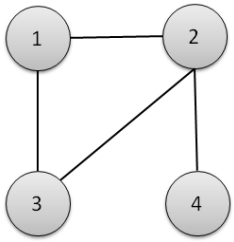

Figura 4-3: Grafo no dirigido

Matemáticamente se pueden realizar diversos algoritmos de grafos que, dependiendo del área de aplicación es el significado que se le dan a los resultados. A nivel del tratamiento de los nodos, se pueden identificar individuos influyentes en una red, con algoritmos como *degree centrality, closeness centrality, betweennes centrality, eigenvector centrality, page rank*, etc. Para

el presente trabajo son de suma importancia este tipo de algoritmos enfocados en los nodos, sobre todo por el tipo de problema de búsqueda léxica.

Destacando el impacto de la teoría de grafos y su aplicación en el procesamiento de lenguaje natural, se presentan algunos trabajos que utilizan medidas de centralidad para diversas tareas en PLN.

- Erkan y Radev [2004], proponen un trabajo basado en grafos para el cómputo de unidades textuales importantes. Se probó la técnica en resúmenes de texto. Los resúmenes de texto extractivos se basan en el concepto de la saliencia en la oración para identificar las oraciones mas importantes en un documento o un conjunto de documentos. Saliencia se define típicamente en términos de la presencia de palabras particularmente importantes o en términos de similitud de un centroide. Los autores establecieron un nuevo enfoque, *LexRank*, para procesar la importancia de una sentencia basada en el concepto de *eigenvector centrality* en una representación de grafos para las oraciones.

- Xie [2005] estudia diferentes medidas de centralidad que son usadas para predecir frases nominales en los resúmenes de artículos científicos.

- El agrupamiento de textos (*text clustering*) es un proceso no supervisado de clasificación de textos y palabras en diferentes grupos. Varios algoritmos usan el modelo de bolsa de palabras para representar textos y clasificar contenidos. El modelo de bolsa de palabras asume que el orden de la palabras no es importante. El trabajo de Iezzi [2012], propone un nuevo método de agrupamiento, considerando las relaciones entre términos y documentos. Se usan medidas de centralidad para valorar la importancia de palabras y textos en un corpus, que permita clasificar los documentos.

4.3.1. Conceptos básicos de grafos

Con la finalidad de entender los conceptos involucrados en cada uno de los algoritmos de centralidad, que son el núcleo principal de la tesis, se abordarán los conceptos fundamentales de la teoría de grafos, ampliamente expuesta por Zuhair *et al.* [2017].

Un grafo es un conjunto de puntos y líneas que los conectan. Es una forma de representar las relaciones entre una colección de objetos.

Los puntos son llamados "vértices" o "nodos" y las líneas son llamados "lados" o "aristas".

Sea un grafo $G = (X, E), x, y \in X$. La distancia de x a y, denotada por $d(x, y)$, es la longitud de camino(x, y). En caso de no existir tal camino en G, entonces $d(x, y) = \infty$. En este caso G es un grafo disconexo. Poniendo a x y y en dos partes del grafo. El diámetro de G, denotado por $diam(G)$, es la distancia entre los dos nodos más lejanos, o $max_{x,y \in X} d(x, y)$.

Un camino es una secuencia de nodos y lados en los que cada lado conecta el vértice anterior y siguiente de la secuencia. De manera más simple; un camino es una lista de enlaces que están conectados secuencialmente para formar una ruta continua. Un camino termina y finaliza en un vértice. Pueden ser, camino simple, camino *euleriano* y circuito.

- **Camino *euleriano*.** Es un recorrido que no pasa sobre el mismo lado más de una vez.
- **Camino simple (*hamiltoniano*).** Es un recorrido que no pasa por el mismo nodo más de una vez.
- **Circuito**. Es un recorrido que comienza y termina en el mismo nodo y que no repite nodos durante su paso.

Un ciclo es un lado que conecta un vértice con sí mismo. Vértices con ningún vecino (grado $= 0$) son llamados aislados. Dos nodos pueden estar conectados con más de un lado. Tales lados son referidos como "paralelos" o "múltiples". Un par ordenado de vértices es también conocido como "arco". Si (x, y) es un arco, entonces x es llamado el nodo inicial y y el nodo terminal.

4.3.1.1. Nodos

Con base en lo establecido por Zuhair *et al.* [2017], mencionan que los nodos son los elementos principales del grafo y comparten las siguientes características:

- En un grafo $G = (X, E)$ y un vértice $x \in X$: La eliminación de x de G significa remover x del conjunto X y remover todos los lados que contienen x. Sin embargo, la eliminación

de un lado es más fácil que un nodo, porque la remoción de un lado requiere únicamente que el lado sea removido de la lista de lados.

- Para un nodo dado x, el número de todos los vértices adyacentes es llamado "grado" y es denotado por $d(x)$. El grado máximo sobre todos los vértices es conocido como el nodo mas alto de G. Nodos adyacentes son llamados vecinos, y el conjunto de vecinos de un nodo x dado es referido como el vecindario de x, denotado por $N(x)$. El conjunto de todos los lados incidentes a un vértice x es denotado por $E(x)$.

- El grado de un nodo es el número de lados incidentes a él. Un nodo aislado es un vértice con grado cero, que es, un nodo que no es punto final de ningún lado. Un nodo hoja es un vértice con grado uno.

- Los dos vértices que están conectados por un lado son llamados sus puntos finales. El lado es descrito como incidente a los vértices. Una simple adyacencia entre dos nodos significa que hay exactamente un lado entre ellos.

- En un grafo dirigido, el grado de salida es el número de lados salientes, mientras que el grado de entrada es el número de lados entrantes. Un nodo fuente es un nodo con grado de entrada en cero, mientras que un nodo sumidero es un nodo con grado de salida en cero.

- Un nodo de corte es un vértice que, si es removido, el número de componentes en el grafo se incrementa. Un conjunto separador es una colección de nodos que si son removidos, el grafo sería desarmado en componentes pequeños.

4.3.1.2. Tipos de grafos

La clasificación presentada en Zuhair *et al.* [2017] identifica la existencia de diferentes tipos de grafos dentro de la teoría de grafos para representar la relación entre los nodos. Los más comunes son:

1. **Grafos no dirigidos.** Un grafo que contiene lados sin dirección. Este tipo de grafo es usado para representar enlaces simétricos.

2. **Grafos dirigidos.** También llamados digrafos, es un grafo que contiene lados con dirección. En este tipo de grafos se pueden presentar relaciones como maestro-alumno o jefe-empleado en las que se asume que la relación es asimétrica.

3. **Grafos con peso.** Grafos teniendo peso de valores reales asociados con los lados. Los pesos en los lados pueden representar conceptos como costos de conexión, longitud, capacidad, similitud, distancia, etc. Los cuales dependen del uso específico de ese grafo. El peso del grafo es calculado como la suma de los pesos asociados en todos los lados.

4. **Grafo plano.** Es un grafo representado en un plano de dos dimensiones sin que ningún lado se cruce. Un ejemplo de su uso, es en el diseño de circuitos eléctricos, que conectan chips, y se requiere que los cables que conectan los componentes no se deben de cruzar entre si.

5. **Grafo ortogonal.** Un grafo teniendo líneas horizontales y verticales.

También existen otro tipo de grafos que despliegan los datos de diferentes formas:

1. **Grafo simple.** Es un grafo no dirigido que no contiene ciclos, o multi-lados conectando cualesquiera dos nodos. Cada lado conecta un par de nodos distintos.

2. **Grafo regular.** Un grafo regular (uniforme) es un grafo en el cual, todos los vértices tienen el mismo número de vecinos, que es el grado de cada vértice.

3. **Grafo completo.** Un grafo en el que cualquier par de nodos están conectados (usualmente escritos como K1, K2,...).

4. **Grafo mixto.** Es un grafo en el cual algunos lados son dirigidos y otros son no dirigidos.

5. **Multigrafo.** Es un grafo que puede tener lados múltiples; permitiendo relacionar los mismos nodos. También pueden contener ciclos.

6. **Grafos con lados sueltos.** Es un grafo con lados que únicamente tienen un final, que son llamados mitad de lado o lados sueltos. Algunos ejemplos son grafos de signos y grafos de sesgo.

7. **Grafos finitos e infinitos.** Un grafo finito es un grafo $G(V, E)$ en el cual V y E son conjuntos finitos. Un grafo infinito es un grafo con un conjunto infinito de nodos, lados, o ambos.

8. **Grafo conexo y disconexo.** Un grafo es llamado conexo si cada par de nodos distintos en este grafo está conectado; de lo contrario, es llamado disconexo.

9. **Grafo k-nodos conectado.** Un grafo es llamado k-nodos conectado o k-lados conectado si no hay $k-1$ nodos (o lados, respectivamente) desconectando el grafo; es decir, permanece conectado siempre que se eliminen menos de k nodos. Un grafo k-nodo conectado es llamado simplemente k-conectado.

10. **Grafo débilmente y fuertemente conectado.** Un grafo dirigido es llamado débilmente conectado si al reemplazar todos los lados dirigidos con lados no dirigidos produce un grafo conexo (no dirigido) . Un grafo es fuertemente conectado (o fuerte) si contiene un camino dirigido de u a v y un camino dirigido de v a u, para cada par de nodos u y v.

4.3.1.3. Formas de representación de grafos

Retomando los conceptos básicos de la teoría de grafos, expuestos por Zuhair *et al.* [2017], las principales formas de representaciones de grafos son las siguientes. Una matriz es una forma de representar y resumir los datos de un grafo. Es un arreglo de elementos que contienen la misma información que un grafo. Tanto los grafos como las operaciones matriciales han sido fundamentales en la construcción de conceptos en el análisis de redes sociales. Sin embargo, las matrices son más apropiadas para computación y el análisis computacional. En general, una matriz es una colección de filas y columnas.

Matriz de adyacencia. La matriz de adyacencia es una matriz con filas y columnas representando los nodos, donde el elemento A_{ij} muestra el número de enlaces que van de i a j (se vuelve simétrica para grafos no dirigidos). Para grafos dirigidos, la matriz de adyacencia toma la siguiente forma:

$$\begin{array}{c} \\ \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

El valor (1) denota la presencia de una relación (lado) entre un nodo A y un nodo B, mientras que el valor (0) significa que no existe tal relación. La mayor desventaja de las matrices de adyacencia es que las celdas con valor igual a cero ocupan la misma cantidad de memoria que las otras celdas.

Lista de lados. Las listas de lados nacieron como una solución al problema de esparcimiento en las matrices de adyacencia. En este tipo de representaciones, se necesita la lista de vértices en la que cada lado es incidente. Sin embargo, en este tipo de representación de los datos, el tamaño puede crecer linealmente con el número de lados.

Listas de adyacencia. A pesar de que las listas de nodos mitigaron significativamente el problema de la memoria reducida y el esparcimiento de datos, no son los suficientemente rápidas para búsquedas o recorridos en los grafos. Para dar una mejor solución, se introdujeron las listas de adyacencia: una lista de enlaces en la que los elementos "$i \rightarrow j$" muestra un lado desde el nodo i al nodo j (también representado como "$i \rightarrow (\text{j}_1, j_2, j_3, ...)$". En este tipo de representación, la búsqueda es rápida. Asimismo, añadir o eliminar nodos/lados es fácil.

4.3.1.4. Recorridos de grafos

Siguiendo los conceptos de Zuhair *et al.* [2017], la mayoría de las herramientas de análisis iteran sobre los nodos y lados del grafo, calculando alguna cantidad de interés. El objetivo puede ser encontrar el camino más corto entre dos puntos, encontrar grupos (*clusters*) conectados, o calcular la centralidad de nodos y lados. En algunos casos, el objetivo es recorrer el grafo entero para entenderlo o recolectar datos de la red. En cualquier caso, estas técnicas requieren de la implementación de uno de los algoritmos de recorrido que fueron diseñados para ya sea encontrar el camino más corto entre dos puntos o recorrer el grafo y tratar de entender su estructura.

Recorrido primero en profundidad. Este algoritmo también es conocido como *Depth-First Traversal o DFS* por sus siglas en inglés. Es una técnica de búsqueda no informada que sistemáticamente recorre el grafo hasta encontrar su objetivo. El algoritmo recorre hacia abajo tomando los hijos de los hijos y después retrocede hacia cada uno de los hermanos, lo cual finalmente produce un árbol de expansión de los nodos que ha visitado.

Recorrido primero en anchura. También conocido como *Breadth-First Traversal o BFT* por sus siglas en inglés. El algoritmo consiste en iterar sobre los vecinos de todos los nodos del vecindario en turno y añadirlo a todos los siguientes vecindarios que no han sido visitados. El algoritmo también puede descubrir, con cambios menores, si existen más de un camino más corto entre dos nodos.

La forma en la que el algoritmo primero en anchura es aplicado para encontrar los caminos más cortos, es de la manera siguiente: partiendo de un nodo “s” que tiene distancia cero hacia sí mismo, mientras que las distancias hacia todos los demás nodos son desconocidas. Después, se encuentran todos los vecinos de “s” que por definición tienen distancia uno desde “s”, posteriormente se toman los vecinos de estos nodos. Excluyendo aquellos que ya han sido visitados, estos vértices deben tener distancia dos desde “s” y sus vecinos, excepto aquellos que ya fueron visitados, tienen distancia 3 y así sucesivamente.

4.3.1.5. Algoritmo de Dijkstra

Este algoritmo fue publicado por Edsger Dijkstra en 1959 y es uno de los algoritmos más importantes en redes de comunicación moderna y es la base de muchos algoritmos de enrutamiento conocidos que son usados en Internet. El algoritmo de Dijkstra encuentra el camino más corto desde un nodo dado hacia cualquier otro nodo en el mismo grafo, por medio de la longitud de sus lados. Esto se realiza manteniendo un registro de los caminos más cortos que el algoritmo ha encontrado hasta ese momento, mientras que este registro es actualizado cada vez que un nuevo camino más corto es encontrado. Al final de la implementación, la distancia más corta a cada nodo es determinada.

4.3.2. Medidas de centralidad

En todos los tipos de redes, los nodos usualmente no son independientes unos de otros. Más bien, están conectados por uno o más lados. Ya que los nodos están conectados pueden estar influenciados entre sí. Las medidas de centralidad ayudan a identificar los nodos importantes, dentro de toda la gran cantidad de ellos que pertenecen al grafo. Estas medidas tienen la habilidad de capturar la importancia de los nodos desde perspectivas diferentes. La centralidad es un término para definir la importancia de un nodo. Centralidad es una de las medidas más comunes para determinar que tan central es un nodo en un grafo.

A continuación se presentan las medidas de centralidad que se usaron para construir el modelo de búsqueda léxica. Se establecen los aspectos teóricos de cada medida.

4.3.2.1. Centralidad Intermedia *(Betweenness centrality)*

El algoritmo de Centralidad Intermedia (CI) presentado por Freeman [1977] se basa en la idea de que un nodo es más importante si intermediario en el grafo [Zuhair *et al.*, 2017]. Para un nodo dado v en un grafo G, la CI es calculada como la relación entre el número de caminos más cortos entre los nodos i y j que pasan a través del nodo v y el número de caminos más cortos entre los nodos i y j. Se describe formalmente de la manera siguiente:

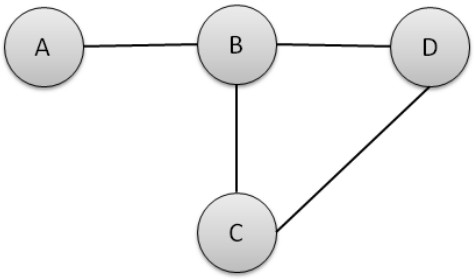

Figura 4-4: Grafo no dirigido para cálculo de CI

$$C_{CI}(v) = \sum_{i,j \in V} \frac{\sigma_{i,j}(v)}{\sigma_{i,j}} \tag{4-2}$$

donde:

$V =$ es el conjunto de nodos, $\sigma_{i,j}$ es el número de caminos más cortos entre i y j, y $\sigma_{i,j}(v)$ es el número de esos caminos que pasan a través del nodo v que no es i o j.

La figura 4.3.2.1 muestra un grafo no dirigido sin pesos, para el cálculo de CI se considera, en este caso, que cada lado tiene un peso constante de 1. Se desglosa de manera detallada la obtención de la CI para el nodo B. Este cálculo se ejecutaría de manera similar en todos los nodos del grafo, obteniendo la CI para cada nodo.

$$\begin{aligned} C_{CI}(B) &= \frac{\sigma_{A,B}(B)}{\sigma_{A,B}} + \frac{\sigma_{A,C}(B)}{\sigma_{A,C}} + \frac{\sigma_{A,D}(B)}{\sigma_{A,D}} + \frac{\sigma_{B,C}(B)}{\sigma_{B,C}} + \frac{\sigma_{B,D}(B)}{\sigma_{B,D}} + \frac{\sigma_{C,D}(B)}{\sigma_{C,D}} \\ &= \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{0}{1} \\ &= 5 \end{aligned} \tag{4-3}$$

En un grafo sin pesos, el algoritmo busca el camino más corto. En un grafo con peso, como el que se ha construido tomando los pesos de las NAP en los lados del grafo, se busca el camino que minimiza la suma de los pesos en los lados.

El algoritmo CI fue introducido teniendo como idea general que cuando una persona en un grupo está estratégicamente localizada en el camino más corto de comunicación respecto a otros pares, la persona está en una posición central [Bavelas, 2002].

Para el desarrollo del modelo de búsqueda léxica basado en CI, se escogió una variación del algoritmo centralidad intermedia, el cual en lugar de procesar la CI de todos los pares de nodos en un grafo, calcula la centralidad basada en una muestra (subconjunto) de nodos [Brandes, 2008].

4.3.2.2. Comunicabilidad de Centralidad Intermedia *(Communicability Betweenness centrality)*

La Centralidad Intermedia ha sido definida considerando la transmisión de información solamente a través de los caminos más cortos entre dos nodos. La Comunicabilidad de Centralidad Intermedia (CCI), se define considerando cualquier recorrido sin importar su longitud, sin embargo, introduce un escalamiento en el que los caminos de mayor longitud, son menos importantes [Estrada *et al.*, 2009]. La CCI muestra buenos resultados al recuperar información significativa en redes de datos, que a pesar de ser de otras áreas de aplicación, puede ser un algoritmo adecuado como motor de la búsqueda léxica.

Esta medida de centralidad argumenta que la mayoría de la información podría fluir por caminos que no son los más cortos. Tomando en cuenta esto, Freeman *et al.* [1991] introdujeron el concepto de Flujo de Centralidad Intermedia (FCI) o *flow betweenness centrality*, que contiene elementos de ambas aproximaciones. Por un lado, considera los caminos más cortos y por otro, los caminos sin esta característica. Sin embargo, Newman [2005] mostró que esta medida en algunas ocasiones puede proporcionar resultados que no son del todo intuitivos, y ofreció una alternativa llamada Caminos Aleatorios de Centralidad Intermedia (CACI) o *random walk betweenness centrality.*

El valor de CACI de un nodo r es igual al número de veces que un camino aleatorio que empieza y termina en otros dos nodos p y q pasan a través del nodo r a lo largo de camino, promediado sobre todos los p y q. Esta medida tiene una idea opuesta la CI basada en caminos más cortos. En la centralidad de los caminos más cortos, la información que se transmite de p a q pasando por r conoce la forma más efectiva para llegar a su objetivo. En la centralidad de Newman se asume que la información no conoce nada de la mejor ruta para llegar a su

objetivo. Se viaja de p a q siguiendo lados de manera aleatoria. En el supuesto que el nodo r se visita muchas veces, sería un indicio que el nodo r juega un papel importante en la transmisión del mesaje de p a q. Promediando todos los nodos p y q se produce una medida general de centralidad sobre el nodo r.

Las dos aproximaciones son diametralmente opuestas, pero cada una tiene un rol válido en la contabilidad de los procesos en la red. Se considera que la CCI es el punto medio entre estas dos medidas, ya que es una medida de centralidad intermedia basada en todos los caminos conectando dos nodos p y q que pasan por el nodo r pero ponderado por la longitud del camino, es decir, una centralidad intermedia que contabilice los caminos más cortos que conectan dos nodos, pero que también reconoce la existencia de otros caminos que no son los más cortos, dándoles menos importancia. La idea de usar sumas ponderadas sobre los caminos de cualquier longitud ha sido usada exitosamente para definir la comunicabilidad entre pares de nodos [Estrada y Hatano, 2008].

En el trabajo de Estrada y Hatano se introdujo la medida de comunicabilidad de la siguiente manera. Si $P_{pq}^{(s)}$ es el número de caminos más cortos entre nodos distintos p y q, teniendo una longitud s y siendo $W_{pq}^{(k)}$ el número de caminos conectando p y q de longitud $k > s$, se considera la cantidad

$$G_{pq} = \frac{1}{s!}P_{pq}^{(s)} + \sum_{k>s}\frac{1}{k!}W_{pq}^{(k)} \tag{4-4}$$

La medida de comunicabilidad entre los nodos p y q. Esta expresión se puede escribir como la suma de las entradas p,q de las diferentes potencias de la matriz de adyacencia. Esto se debe a que cada elemento de la matriz de adyacencia elevada a una potencia *k-ésima* representa el número de caminos existentes de longitug k, entre los nodos *p-ésimo* (representando la fila) y el *q-ésimo* (representado la columna). De tal forma que, G_{pq} se puede escribir

$$G_{pq} = \sum_{k=0}^{\infty}\frac{(A^k)_{pq}}{k!}, \tag{4-5}$$

la cual converge a

$$G_{pq} = (e^A)_{pq}. \tag{4-6}$$

Usando esta aproximación, Estrada *et al.* [2009] definieron la centralidad intermedia con base en la función de comunicabilidad. Considerando la fórmula descrita en 4-4 se definió G_{pqr} como la suma con peso, donde solo se consideran caminos que contengan a r. Siendo así, la nueva fórmula para definir la CCI de un nodo r es

$$w_r = \frac{1}{C} \sum_p \sum_q \frac{G_{prq}}{G_{pq}}, p \neq q, p \neq r, q \neq r, \tag{4-7}$$

donde $C = (n-1)^2 - (n-1)$ es un factor de normalización igual al número de términos de la suma, tal que w_r toma valores entre cero y uno.

Bajo esta nueva perspectiva, es posible encontrar aquellos nodos más influyentes en la red y, en una visión más completa, se toma en cuenta cualquier camino en el grafo.

4.3.2.3. Centralidad de Cercanía *(Closeness Centrality)*

La Centralidad de Cercanía (CC) se calcula considerando los nodos que tienen el promedio más pequeño de la longitud del camino más corto. La CC es importante porque toma en cuenta no solamente las conexiones inmediatas de un nodo, también considera las conexiones indirectas de los otros nodos del grafo. Es una medida de riqueza, que identifica la rapidez de esparcimiento de la información a los otros nodos desde un nodo específico [Al-Taie y Kadry, 2017]. En esta forma de centralidad, un nodo es más importante si puede alcanzar todos los otros nodos en el menor número de pasos [Brath y Jonker, 2015]. En una red social, esta medida podría ayudar a identificar aquellos nodos más influyentes sobre los que las comunicaciones pasan a través de algunos intermediarios (por ejemplo, un asistente), pero actúa como un puente entre diferentes grupos.

La CC de un nodo u, propuesta por Freeman [1979], es el recíproco del promedio de los caminos más cortos a v sobre todos los *n-1* nodos alcanzables.

$$CC(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)}, \tag{4-8}$$

donde *d(v,u)* es el camino más corto entre v y u, y n es el número de nodos que puede alcanzar u. Los valores más altos de cercanía, indican centralidad mayor.

Posteriormente, Wasserman y Faust [1994] propusieron una mejora de esta medida para grafos con más de un componente conectado. El resultado es una relación de la fracción de nodos del grupo a los que se puede llegar, con la distancia promedio de los nodos accesibles.

$$CC_{WF}(u) = \frac{n-1}{N-1}\frac{n-1}{\sum_{v=1}^{n-1} d(v,u)}, \tag{4-9}$$

donde N es número de nodos en el grafo.

Este algoritmo, al trabajar con caminos más cortos, permite al igual que con la Centralidad Intermedia, poder probar cual de las 3 diferentes funciones de pesado: tiempo, frecuencia inversa o fuerza de asociación inversa, proporciona la mejor precisión de la búsqueda léxica. Asimismo, es posible probar el comportamiento de las variantes que posee esta medida de centralidad. Ya sea la de Freeman o la de Wasserman y Faust

4.3.2.4. Centralidad Katz *(Katz Centrality)*

La Centralidad Katz (CK) o *Katz Centrality* [Katz, 1953] suma todas las distancias ponderadas a todos los demás nodos desde un nodo dado. Cuanto más alejado esté un nodo del nodo medido, menor será el peso y menor será la contribución a la centralidad [Brath y Jonker, 2015].

Para el cómputo de la CK en un nodo, se calcula la centralidad de sus vecinos. Es una generalización de la Centralidad por Eigenvectores. La CK de un nodo i es

$$x_i = \alpha \sum_j A_{ij} x_j + \beta, \tag{4-10}$$

donde A es la matriz de adyacencia del grafo G con eigenvectores λ. El parámetro β controla

la centralidad inicial y

$$\alpha < \frac{1}{\lambda_{\text{máx}}}. \tag{4-11}$$

La Centralidad Katz calcula la influencia relativa de un nodo dentro de un grafo, midiendo el número de vecinos inmediatos y, también todos los demás nodos que se conectan al nodo a través de estos vecinos inmediatos. Se puede proporcionar peso adicional a los vecinos inmediatos a través del parámetro β. Sin embargo, las conexiones realizadas con vecinos distantes son penalizadas por un factor de atenuación α que debe ser estrictamente menor que el valor propio inverso más grande de la matriz de adyacencia para que la CK se calcule correctamente [Newman, 2010].

El algoritmo utiliza el método de potencia para encontrar el vector propio correspondiente al valor propio más grande de la matriz de adyacencia de G. El parámetro α debe ser estrictamente menor que el inverso del valor propio más grande de la matriz de adyacencia para que el algoritmo converja.

Cuando $\alpha = 1/\lambda_{\text{máx}}$ y $\beta = 0$, la CK es la misma que la Centralidad de Vector Propio. Permite que un nodo que tiene muchos vecinos tenga alta centralidad independientemente de si esos vecinos tienen una alta centralidad, y esto podría ser deseable en algunas aplicaciones.

Este algoritmo, también toma como base los caminos más cortos, al igual que con CI y CC. Por lo tanto, al implementarlo usando el corpus NAP se utilizarán las funciones de pesado: tiempo, frecuencia inversa y fuerza de asociación inversa.

4.3.2.5. Centralidad de Vector Propio *(Eigenvector centrality)*

La Centralidad de Vector Propio (CVP) [Bonacich, 1987] es proporcional a la suma de la centralidad de los vecinos. Es una medida de influencia relacionada, identifica quien está más cerca de los nodos más influyentes en el grafo. Un nodo es importante si está conectado a otros nodos. Esto muestra que un nodo con un número pequeño de contactos influyentes puede superar aquellos con un número mayor de contactos mediocres. En otras palabras, las personas

bien conectadas valen más que las personas mal conectadas [Zuhair *et al.*, 2017].

La CVP es similar a la Centralidad Katz, pero es una aproximación recursiva donde un nodo es más probable que sea central si sus vecinos son centrales [Brath y Jonker, 2015]. La Centralidad por Vector propio para un nodo i es el *i-ésimo* elemento del vector x definido por la ecuación

$$Ax = \lambda x. \tag{4-12}$$

donde A es la matriz de adyacencia para el grafo G con el valor propio λ. De acuerdo con el teorema de Perron-Frobenius [Pillai *et al.*, 2005] existe una única solución x, si λ es el valor propio más grande de la matriz de adyacencia A.

La CVP permite trabajar con el grafo pesado, al igual que con algunos de los algoritmos anteriores. Por lo tanto, es posible probar cual de las tres diferentes funciones de pesado: tiempo, frecuencia inversa o fuerza de asociación inversa, proporciona los mejores resultados en la búsqueda léxica. Dentro de las posibilidades de esta implementación, también es posible determinar el comportamiento cuando los pesos de los lados en el grafo es constante.

4.3.2.6. Centralidad de Cercanía y Flujo Corriente *(Current Flow Closeness Centrality)*

La Centralidad de Cercanía y Flujo Corriente (CCFC) es una variación de Centralidad Intermedia y Centralidad de Cercanía, con un modelo diferente de transmisión de información. En lugar de tomar solamente los caminos más cortos, se asume que la información se distribuye eficientemente como en una corriente eléctrica.

Una crítica común para las medidas basadas en caminos más cortos, es que no toman en cuenta la distribución de información a través de caminos que no sean los más cortos y por lo tanto, no son apropiados para los casos donde la distribución de los lados está gobernada por otro tipo de reglas [Brandes y Fleischer, 2005]. La medida CCFC considera grafos $G = (V, E)$ que sean simples, no dirigidos, conectados y con vértices $n \geq 3$. Una red eléctrica $N = (G; c)$ es un grafo con pesos positivos en los lados $c : E \to \mathbb{R}_{>0}$ indicando la conductividad o fuerza

de un lado. Equivalentemente, la red puede ser definida en términos de pesos positivos $r : E \rightarrow \mathbb{R}_{>0}$ indicando la resistencia o longitud de un lado, donde la conducción y resistencia están relacionados por $c(e) = 1/r(e), \forall e \in E$.

La CCFC está basada en cómo la corriente fluye por la red, considerando leyes aplicables a este tipo de redes eléctricas. Esta métrica también es conocida como Centralidad de Información *(Information Centrality)* [Stephenson y Zelen, 1989].

Como se mencionó anteriormente, la CCFC trabaja con grafos pesados, que simulan la resistencia de una red eléctrica. En cuyo caso, se probaran las diferentes funciones de pesado en NAP: tiempo, frecuencia inversa o fuerza de asociación inversa. Asimismo, esta implementación puede ser probada cuando el grafo no tiene peso en los lados, en cuyo caso se considera un peso constante.

4.3.2.7. Centralidad de Grado *(Degree Centrality)*

La Centralidad de Grado (CG) es la más simple. Considera el nodo con el grado más alto (mayor número de conexiones) como el nodo mas importante del grafo. La CG se enfoca en nodos individuales, simplemente cuenta el número de lados que un nodo tiene. Es una medida de popularidad que determina los nodos que pueden distribuir información a un área localizada. [Zuhair *et al.*, 2017]. En una red social como Facebook o Twitter, es una persona muy bien conectada, y la importancia de esa persona radica en que probablemente sabrá lo que está sucediendo a su alrededor porque él o ella, están muy bien conectados [Brath y Jonker, 2015].

Capítulo 5

Modelo de búsqueda léxica basado en grafos

Este capitulo presenta el modelo creado para la búsqueda léxica. Como base para la implementación de este proceso, se construyó un grafo que contiene por nodos cada una de las palabras del Corpus de Normas de Asociación de Palabras (NAP). Posteriormente, se ejecutan algoritmos de búsqueda sobre el grafo creado, utilizando las palabras de una definición dada.

Las palabras que serán devueltas por los algoritmos, corresponden a las palabras objetivo. Para este propósito, se consideraron las medidas de centralidad como: centralidad de grado, centralidad de cercanía, centralidad intermedia, comunicabilidad de centralidad intermedia, centralidad Katz, centralidad de vector propio y centralidad de cercanía y flujo corriente. Como se abordó en el capítulo anterior, las medidas de centralidad identifican los nodos más importantes en el grafo, cada una dando énfasis a criterios particulares.

Como se mencionó anteriormente, para la construcción del grafo se usaron las Normas de Asociación de Palabras. Los términos compuestos, por ejemplo Nueva_York, son tratados como una sola palabra. Para las NAP del español de México, se tiene la limitante de que, cada *estímulo* es un sustantivo tangible. Sin embargo, esta es una regla propia de la construcción de ese recurso en particular. Otras NAP existentes, como las Normas de Asociación Libres en Castellano o el *Edinburgh Associative Thesaurus*, no tienen esa característica.

5.1. Representación de NAP con grafos

El grafo representando el NAP ha sido elaborado con los elementos léxicos lematizados. Es formalmente definido como: $G = \{V, E, \phi\}$ donde:

- $V = \{v_i | i = 1, ..., n\}$ es un conjunto finito de nodos de longitud n, $V \neq \emptyset$, que corresponde a los estímulos y sus respuestas.
- $E = \{(v_i, v_j) | v_i, v_j \in V, 1 \leq i, j \leq n\}$, es el conjunto de lados.
- $\phi : E \to \mathbb{R}$, es una función sobre los pesos de los lados.

El grafo es no dirigido, por lo tanto cada estímulo está conectado a cada respuesta sin ningún orden de precedencia.

Para el peso de los lados, existen tres funciones diferentes:

Tiempo (T) Mide cuantos segundos toma a un participante realizar una respuesta para cada *estímulo*.

Frecuencia (F) Cuenta el número de ocurrencias de cada respuesta a su estímulo en todo el corpus. Para el funcionamiento del sistema con los caminos más cortos, se necesitó calcular la FI, frecuencia inversa, que es definida en la siguiente manera: siendo F la frecuencia de una palabra asociada, y ΣF la suma de las frecuencias de las palabras conectadas al mismo estímulo, $FI = \Sigma F - F$.

Fuerza de Asociación (FA) Establece una relación entre la frecuencia (F) y el número de respuestas para cada estímulo. Es calculada de la siguiente forma: siendo F la frecuencia de una palabra dada, y ΣF la suma de las frecuencias de las palabras conectadas al mismo estímulo (el total de número de respuestas), la fuerza de asociación (FA) de la palabra W para tal estímulo es dada por la fórmula:

$$FA_W = \frac{F * 100}{\Sigma F}$$

Para los experimentos, se necesita calcular la fuerza de asociación inversa, FAI, con la finalidad de preparar el sistema para trabajar con algoritmos basados en grafos:

$$FAI_W = 100 - \frac{F * 100}{\Sigma F}$$

5.2. Medidas de centralidad aplicadas a la búsqueda léxica

En cada uno de los algoritmos presentados en esta sección, se indicará si el grafo requiere de una adecuación especial dependiendo del funcionamiento de cada medida de centralidad. Ya que ciertos algoritmos, como la centralidad de grado, calcula la relevancia de un nodo basándose en el número de sus nodos adyacentes, haciendo irrelevante el peso que une dos nodos. En otras ocasiones, el grafo original necesita ser modificado para obtener mejores resultados.

5.2.1. Modelo basado en centralidad intermedia (CI)

En la sección 4.3.2.1 se especificó que la versión adecuada para trabajar el diccionario inverso es la adaptación de centralidad intermedia, que trabaja con subcojuntos de nodos iniciales y finales. Estos sirven para encontrar los caminos más cortos entre ellos. Por lo tanto, se definió un subgrafo compuesto por las palabras (nodos) de una definición. Por ejemplo, si se tiene la siguiente definición:

Animal que vive en el bosque y le gusta comer nueces

Se tomarán las palabras: animal, vivir, bosque, comer y nuez, como los elementos del subconjunto en la CI. Se puede observar que las palabras tomadas para construir el subcojunto no son exactamente las establecidas en la definición, en el Algoritmo 2 se aborda el proceso de transformación de las palabras. Posteriormente, el grupo obtenido es usado como nodos iniciales y finales, para calcular el camino más corto desde cada uno de los nodos del conjunto inicial hacia cada uno de los nodos del conjunto final. Finalmente, los nodos son ordenados con base en la medida de la CI tomada como parámetro de comparación de los nodos más importantes encontrado por el algoritmo.

Algoritmo 2: Búsqueda léxica CI

Datos: Corpus NAP, definiciones a buscar
Resultado: Lista ordenada de conceptos

```
pre-procesar(Corpus NAP);
pre-procesar(definiciones a buscar);
GraphNAP = construir-grafo(Corpus NAP);
para cada definición hacer
    definición = filtrar-PalabrasEnNAP(definición);
    construir_subgrafo(definición);
    nodos_ordenados = CI(GrafoNAP,subgrafo);
fin
```

El algoritmo 2 presenta el esquema general del modelo construido. Primero, se realizan algunos pasos de pre-procesamiento. Todos los estímulos y las respuestas se encuentran lematizados, dejando cada palabra como una unidad léxica sin flexionar. El mismo pre-procesamiento es aplicado a las definiciones que serán buscadas por el modelo. Este proceso proporciona de una mayor coincidencia en el caso de que una definición contenga *mesa*, *mesas*, etc. porque será transformado a su lema, *mesa*. Para este proceso, se usó el proceso de lematización proporcionado por *Freeling* [1] [Padró y Stanilovsky, 2012] para el idioma español.

Posteriormente, se construyó el grafo NAP utilizando el paquete de Python, Networkx [Hagberg *et al.*, 2005], versión 2.4. Después, para cada definición a ser buscada, se eliminan las palabras funcionales usando la lista de *stop words* disponible en el paquete de *NLTK* [Bird y Loper, 2004], versión 3.1. Después, con la lista de palabras con significado léxico, se seleccionaron únicamente aquellas que pertenezcan al vocabulario en NAP. Con esto se construyó un subgrafo que es usado como entrada en el algoritmo de centralidad intermedia. Finalmente, los nodos son ordenados de acuerdo con la medida más alta de la centralidad, que corresponde a las palabras que están más cerca de la definición.

5.2.2. Modelo basado en comunicabilidad de centralidad intermedia (CCI)

La implementación del modelo basado en la comunicabilidad de centralidad intermedia, aplicada a la búsqueda léxica y a las NAP, requiere primero de un proceso de reducción en el

[1] `http://www.corpus.unam.mx/servicio-freeling/`

espacio de búsqueda. Lo importante es obtener la CCI de solamente aquellos nodos que estén relacionados con la consulta onomasiológica en turno. Para ello, se realizó una construcción de un grafo nuevo, más pequeño que las NAP, tomando solo unos cuantos nodos del grafo. Los nodos seleccionados para esta construcción son aquellos que estén conectados con las palabras que forman parte de una consulta, sin descartar estas mismas palabras que ayudaron a encontrar las conexiones. De esta forma, el grafo original es reducido considerablemente y, sobre éste, se ejecuta la CCI. Posteriormente, los nodos son ordenados con base en la medida CCI, este valor es usado como parámetro de comparación para cada uno de los nodos del grafo reducido.

Algoritmo 3: Búsqueda léxica CCI

```
   Datos: Corpus NAP, definiciones a buscar
   Resultado: Lista ordenada de conceptos
1  pre-procesar(Corpus NAP);
2  pre-procesar(definiciones a buscar);
3  GrafoNAP = construir-grafo(Corpus NAP);
4  para cada definición hacer
5  |  definición = filtrar-PalabrasEnNAP(definición);
6  |  Nuevo_Grafo_NAP = construir_nuevo_grafo(GrafoNAP, definición);
7  |  nodos_ordenados = CIC(Nuevo_Grafo_NAP);
8  fin
```

El algoritmo 3 presenta el esquema general del modelo construido. Los pasos de pre-procesamiento, del corpus NAP y las definiciones son los mismos que en el descrito en el algoritmo 2.

Al igual que en la descripción del motor basado en la CI, se construyó el grafo NAP y por cada definición se eliminan las palabras funcionales, conservando solo las que pertenezcan al vocabulario de las NAP.

El nuevo grafo NAP corresponde a una reducción de grafo NAP original. Por ejemplo, si una definición, después de haber preprocesado los tokens, tiene el siguiente conjunto de palabras: miel, insecto, volar, amarillo. El nuevo grafo se construiría verificando el grafo NAP original, para obtener cada uno de los nodos vecinos de estas palabras. En el caso presentado anteriormente, se obtiene el nuevo nuevo grafo con nodos de los vecinos a una distancia 1. Los nuevos nodos podrían ser: dulce, abeja, picar, etc. Todas ellas se añaden al conjunto que sirvió

para realizar la búsqueda original con sus correspondientes lados.

Finalmente, los nodos del nuevo grafo son ordenados de acuerdo con la medida más alta de la CCI, que corresponde a las palabras que están más cerca de la definición. Es importante mencionar que, por cada definición a buscar dentro del modelo, se van generando pequeños grafos sobre los cuales realizar la búsqueda. El tamaño de ellos depende la riqueza léxica presente en cada definición.

5.2.3. Modelo basado en centralidad de cercanía (CC)

Al igual que en la CCI, el modelo basado en centralidad de cercanía requiere para su implementación una reducción del espacio de búsqueda en el grafo NAP. Recordando la metodología de recorte respecto al grafo original, se toman solo unos cuantos nodos y son aquellos que estén conectados con las palabras que forman parte de la consulta, incluyendo estas mismas palabras que ayudaron a encontrar las conexiones. Este grafo final permite realizar la ejecución de la medida CC. Posteriormente, los nodos son ordenados con los resultados arrojados por la medida de la CC, tomada como parámetro de comparación de los nodos más importantes encontrados por el algoritmo.

Algoritmo 4: Búsqueda léxica CC

```
   Datos: Corpus NAP, definiciones a buscar, tipo
   Resultado: Lista ordenada de conceptos
1  pre-procesar(Corpus NAP);
2  pre-procesar(definiciones a buscar);
3  GrafoNAP = construir-grafo(Corpus NAP);
4  para cada definición hacer
5  |  definición = filtrar-PalabrasEnNAP(definición);
6  |  Nuevo_Grafo_NAP = construir_nuevo_grafo(GrafoNAP, definición);
7  |  nodos_ordenados = CC(Nuevo_Grafo_NAP, peso, tipo);
8  fin
```

El algoritmo 4 presenta el esquema general del modelo construido. Los pasos de preprocesamiento, del corpus NAP y las definiciones son los mismos que en el descrito en los algoritmos 2 y 3 .

Al igual que en la descripción del motor basado en la CI y CCI, se construyó el grafo NAP

y por cada definición se eliminan las palabras funcionales, conservando solo las que pertenezcan al vocabulario de las NAP. Todas las palabras se encuentran en su versión lematizada.

El nuevo grafo NAP corresponde a una reducción de grafo NAP original, este nuevo grafo contiene los vecinos de las palabras que sirven de búsqueda, incluyéndolas también en el conjunto; todos ellos a una distancia uno. Una vez encontrados los nodos, se toman del grafo original los diferentes pesos, sobre los cuales se probará el motor basado en CC.

La función CC, visible en la línea 8, recibe además del nuevo grafo, otros dos parámetros: (1) el peso, hace referencia a las tres funciones de pesado disponibles en el corpus NAP, y (2) el tipo que corresponde a la versión clásica de la CC [Freeman, 1979] o la versión mejorada [Wasserman y Faust, 1994].

Finalmente, los nodos del nuevo grafo son ordenados de acuerdo con la medida más alta de la Centralidad de Cercanía, que corresponde a las palabras mas centrales con base en esta medida, que se adecúan mas a la definición probada.

Al igual que en CCI en esta versión del modelo, por cada definición a buscar dentro del modelo, se van generando nuevos grafos y mientras más palabras tenga una definición, mayor será el tamaño del grafo, lo cual vislumbra una mayor posibilidad de precisión en la recuperación del término objetivo.

5.2.4. Modelo de búsqueda basado en centralidad Katz (CK)

Para la implementación de la centralidad Katz aplicada a la búsqueda léxica y a las NAP, se planteó una reducción del espacio de búsqueda como con CI y CC; tomando solo unos cuantos nodos que estén conectados con las palabras que forman parte de la consulta, así como las palabras usadas en búsqueda de las conexiones. Con el grafo resultante, se realiza la ejecución de la medida CK. Finalmente, los nodos son ordenados con los resultados arrojados por el algoritmo CK.

El algoritmo 5 presenta el esquema general del modelo construido para la Centralidad Katz. Los pasos de pre-procesamiento, del corpus NAP y las definiciones son los mismos que en el descrito en los algoritmos 2, 3 y 4.

Algoritmo 5: Búsqueda léxica CK

```
   Datos: Corpus NAP, definiciones a buscar
   Resultado: Lista ordenada de conceptos
1  pre-procesar(Corpus NAP);
2  pre-procesar(definiciones a buscar);
3  GrafoNAP = construir-grafo(Corpus NAP);
4  para cada definición hacer
5  |  definición = filtrar-PalabrasEnNAP(definición);
6  |  Nuevo_Grafo_NAP = construir_nuevo_grafo(GrafoNAP, definición);
7  |  nodos_ordenados = CK(Nuevo_Grafo_NAP,peso);
8  fin
```

Al igual que en la descripción del motor basado en la CI, CCI y CC, se construyó el grafo NAP y por cada definición se eliminan las palabras funcionales, conservando solo las que pertenezcan al vocabulario de las NAP. Todas las palabras se encuentran en su versión lematizada.

El nuevo grafo reducido se construye de la misma manera que como se hizo para el motor de búsqueda basado en CCI y CC.

La función CK, visible en la línea 7 del algoritmo 5, recibe como segundo parámetro el peso, el cual irá tomando los valores ya sea de tiempo, frecuencia o fuerza de asociación inversa.

Finalmente, los nodos del nuevo grafo son ordenados de acuerdo con la medida más alta de la Centralidad Katz, que corresponde a las palabras mas centrales, permitiéndonos identificar aquellas que se adecúan más a la definición probada.

El tamaño del grafo nuevo en cada caso varía, dependiendo de las palabras presentes en cada definición buscada.

5.2.5. Modelo de búsqueda léxica basado en centralidad de vector propio (CVP)

El modelo de búsqueda léxica basado en la centralidad de vector propio, requiere también de una reducción del tamaño del grafo. La metodología para ello es la misma que los recortes hechos en los motores de búsqueda anteriores. Con el grafo reducido, se ejecuta la CVP. En el último paso, los nodos son presentados, ordenando los resultados de la medida de la CVP para

cada uno de los nodos del grafo.

Algoritmo 6: Búsqueda léxica CPV

```
   Datos: Corpus NAP, definiciones a buscar
   Resultado: Lista ordenada de conceptos
1  pre-procesar(Corpus NAP);
2  pre-procesar(definiciones a buscar);
3  GrafoNAP = construir-grafo(Corpus NAP);
4  para cada definición hacer
5  |  definición = filtrar-PalabrasEnNAP(definición);
6  |  Nuevo_Grafo_NAP = construir_nuevo_grafo(GrafoNAP, definición);
7  |  nodos_ordenados = CPV(Nuevo_Grafo_NAP,peso);
8  fin
```

El algoritmo 6 presenta el esquema general del modelo construido. Los pasos de pre-procesamiento, del corpus NAP y las definiciones son los mismos que en el descrito en los algoritmos anteriores.

Se construyó el grafo NAP y por cada definición se eliminan las palabras vacías. Un filtrado adicional es realizado, para conservar solo las palabras que pertenezcan al vocabulario de las NAP. Es importante recordar que todas las palabras están lematizadas.

El nuevo grafo NAP, corresponde a una reducción de grafo NAP original, como se ha explicado anteriormente.

En la línea 7 se observa la llamada a la función CPV. Como primer parámetro se tiene el grafo del que se regresará el conjunto de nodos, cada uno de ellos con la medida de centralidad. El siguiente parámetro es el peso, que puede variar con base en las funciones de pesado disponibles en el corpus NAP.

Finalmente, los nodos del nuevo grafo son ordenados de acuerdo con la medida más alta de la Centralidad de Vector Propio, que corresponde a las palabras mas centrales con base en esta medida, que se adecúan a la definición probada.

5.2.6. Modelo de búsqueda basado en centralidad de cercanía y flujo corriente (CCFC)

La Centralidad de cercanía y flujo corriente aplicada a la búsqueda léxica, y usando las NAP como corpus de entrada, requiere de la reducción del tamaño del grafo, la técnica aplicada para esta disminución ha sido explicada anteriormente. Posteriormente, con el grafo reducido se ejecuta la CCFC. Finalmente, los nodos son ordenados usando la medida de la CCFC como parámetro de comparación.

Algoritmo 7: Búsqueda léxica CCFC

```
   Datos: Corpus NAP, definiciones a buscar
   Resultado: Lista ordenada de conceptos
1  pre-procesar(Corpus NAP);
2  pre-procesar(definiciones a buscar);
3  GrafoNAP = construir-grafo(Corpus NAP);
4  para cada definición hacer
5  |  definición = filtrar-PalabrasEnNAP(definición);
6  |  Nuevo_Grafo_NAP = construir_nuevo_grafo(GrafoNAP, definición);
7  |  nodos_ordenados = CCFC(Nuevo_Grafo_NAP,peso);
8  fin
```

El algoritmo 7 presenta el esquema general del modelo construido para CCFC. Los pasos de pre-procesamiento, del corpus NAP y las definiciones son los mismos que en el descrito en los algoritmos anteriores. Esta misma técnica sucede con la construcción del grafo basado en NAP.

La reducción del grafo original produce un pequeño grafo con los vecinos a inmediatos, es decir, aquellos que se encuentren a distancia uno.

La parte principal de esta versión del motor es visible en la línea 7 en la que se llama a la función CCFC. Los dos parámetros con los que esta función se ejecuta refieren al grafo reducido y a la función de pesado.

Como último paso, se presentan los resultados ordenándolos con base en la centralidad CCFC.

5.2.7. Modelo basado en centralidad de grado (CG)

La implementación del motor de búsqueda léxica usando centralidad de grado, toma como base el corpus NAP. Sobre todo en esta medida tan sencilla, es muy importante hacer la reducción del grafo. De este grafo reducido, la CG identificará aquellos nodos con las mayores conexiones para el cálculo de la medida. Para este fin, la reducción es la misma descrita en los motores anteriores.

El resultado del motor presenta los nodos ordenados, mostrando primero aquellos nodos con la mayor cantidad de vecinos. Para esta medida de centralidad, el peso de los lados del grafo no es relevante.

Algoritmo 8: Búsqueda léxica CG

```
   Datos: Corpus NAP, definiciones a buscar
   Resultado: Lista ordenada de conceptos
1  pre-procesar(Corpus NAP);
2  pre-procesar(definiciones a buscar);
3  GrafoNAP = construir-grafo(Corpus NAP);
4  para cada definición hacer
5  |  definición = filtrar-PalabrasEnNAP(definición);
6  |  Nuevo_Grafo_NAP = construir_nuevo_grafo(GrafoNAP, definición);
7  |  nodos_ordenados = CG(Nuevo_Grafo_NAP);
8  fin
```

El algoritmo 8, presenta el esquema general del modelo construido para CG. Los pasos de pre-procesamiento, del corpus NAP y las definiciones son los mismos que en el descrito en los algoritmos anteriores. Así como la construcción del grafo basado en NAP.

Para la obtención del nuevo grafo reducido, se toman las palabra conectadas a cada uno de los *tokens* de una definición de prueba. Cuando el algoritmo CG es llamado, únicamente recibe como parámetro el grafo reducido.

Finalmente el Algoritmo 8 devuelve la lista ordenada de conceptos basada en la CG.

Capítulo 6

Experimentación y evaluación

Este capítulo presenta los experimentos realizados para evaluar cada uno de los modelos de búsqueda léxica descritos en el capítulo anterior. Asimismo, se presenta el corpus de evaluación recolectado para tal fin. Los modelos también son comparados con algoritmos de recuperación de información para poder medir su efectividad respecto a modelos clásicos de PLN.

Durante el desarrollo de los experimentos sobre las NAP, se observó que, adicionalmente al desarrollo de modelos de búsqueda léxica, y gracias a la estructuración del corpus en formato de grafo, se pueden construir vectores de palabras *(embeddings)* utilizando el algoritmo *node2vec*, que será descrito en este capítulo.

Los experimentos se presentan también en inglés para mostrar la transferibilidad de la tecnología presentada en esta tesis.

6.1. Corpus de evaluación de la búsqueda léxica

El corpus fue reunido con la colaboración de estudiantes que dieron su propia descripción de la palabra. Es importante notar que esta tarea es casi equivalente a la de proporcionar palabras clave.

No se usaron definiciones tomadas de diccionarios porque tienden a ser más científicas y en algunos casos pueden contener tecnicismos, no son del tipo de pistas que un humano en

la búsqueda de una palabra tiende a dar. Sin embargo, el uso de definiciones canónicas de diccionarios comunes podría ser una tarea posible.

6.1.1. Español

Se recolectó un corpus pequeño para evaluar el modelo de búsqueda léxica, conteniendo 5 definiciones diferentes de 56 conceptos correspondientes a estímulos del NAP, con un número total de 280 definiciones. Para ello, se utilizaron formularios en línea y la instrucción presentada para llenar los datos fue la siguiente:

> Suponiendo que tienes una palabra en la punta de la lengua y deseas que alguien más te ayude a encontrar lo que quieres decir. ¿Qué frase le dirías para que te ayude a encontrarla?

Como se puede observar en la instrucción, la recolección no estaba restringida y algunas de las contribuciones de los estudiantes fueron listas de palabras. La lista completa de las descripciones puede ser consultada en el anexo A.

Todas las palabras definidas por los participantes son sustantivos, pero las palabras utilizadas para las definiciones eran libres y no estaban establecidas para alguna categoría gramatical.

La tabla 6-1 presenta un ejemplo de 5 definiciones de los mismos conceptos dados por diferentes estudiantes.

6.1.2. Inglés

Se utilizó un corpus pequeño que contiene 10 definiciones para 7 palabras, estas definiciones fueron tomadas de Sierra y McNaught [2000] que fueron usadas para evaluar su trabajo. Para la recolección de estas definiciones, se reportó en el trabajo orginal, que fueron obtenidas de un grupo de 20 estudiantes de pregrado en el área de terminología. De dos conjuntos, a cada estudiante se le pidió seleccionar un grupo y debía escribir en una hoja en blanco, de manera similar a una búsqueda onomasiológica, una definición o las ideas sugeridas para cada palabra. Después se intercambiaron las hojas con los otros estudiantes que participaron en el experimento

Tabla 6-1: Ejemplos de definiciones de león y queso, dadas por estudiantes.

Concepto	**León**
Definición 1	Ruge y vive en la selva
Definición 2	Rey
Definición 3	Animal carnívoro, de cuatro patas, grande melena, pelaje amarillo. Es el rey de la selva
Definición 4	El animal del escudo de Gryffindor
Definición 5	Animal conocido como el rey de la selva
Concepto	**Queso**
Definición 1	Alimento elaborado con leche. Existen diferentes tipos: manchego, cotija, panela entre otros
Definición 2	El producto que se saca de la leche de la vaca
Definición 3	Amarillo y con agujeros
Definición 4	Derivado lácteo que ponen en trampas para ratones
Definición 5	Como la crema pero sólido

y escribieron la palabra o palabras que identificaban los definiciones escritas en la páginas por el estudiante anterior.

Las palabras seleccionadas para evaluar la metodología son: *water, squirrel, bench, hurricane, lemon, bucket* y *clothes*. Las definiciones de estas siete palabras están disponibles en el Anexo A. La tabla 6-2 presenta un ejemplo de 10 definiciones del mismo concepto dado por diferentes estudiantes.

6.2. Resultados de los modelos de búsqueda léxica

Los experimentos fueron realizados tomando en cuenta grafos con pesos, con las 3 funciones mencionadas anteriormente: Tiempo (T), Frecuencia Inversa (FI) y Fuerza de Asociación Inversa (FAI). En algunos casos, se hicieron pruebas con grafos sin peso para poder hacer una comparativa.

Para la evaluación del proceso de inferencia, se usó la técnica de precisión en k ($p@k$) [Manning *et al.*, 2009]; por ejemplo, $p@1$ muestra que el concepto asociado a una definición dada

Tabla 6-2: Ejemplos de definiciones de *squirrel* dada por los estudiantes.

It's a little rodent and can be red or grey, it has a big bushy tail
A small rodent living in trees with a long bushy tail
A small rodent which lives in trees, collects nuts and has a bushy tail
Animal, grey/red, bushy tail, lives in trees, buries nuts
Small animal, lives in trees, eats acorns, has a bushy tail
Animal, bushy tail, eats nuts, builds nests in trees called dreys
Small funny animal with big, bushy tail, likes nuts, likes trees
Animal that lives in trees and collects acorns, has a long tail
A small-sized animal, habitat in trees
Small grey mammal, relative to the rodent, found in both countryside and town

fue correctamente devuelto en el primer lugar, en $p@3$ el concepto estaba ubicado entre los tres primeros resultados, y lo mismo aplica para $p@5$.

En el repositorio digital *GitHub*[1] se encuentra la relación detallada de los resultados arrojados por el modelo de búsqueda léxica. Los datos mostrados presentan de primera instancia el concepto a buscar, después se tiene la definición dada por los usuarios. Posteriormente, se muestra la relación de palabras devueltas por el algoritmo, indicando la función de peso usada en el grafo. Se tiene un archivo de resultados por cada modelo, conservando la misma organización en todos ellos. La estructura es la siguiente:

Concepto buscado: **Gelatina.**

Definición del usuario: Postre elaborado con agua, puede ser de diversos sabores. Este postre es recomendado para que los enfermos lo consuman.

- *Frecuencia.* ('gelatina', 8.322112351845594e-07), ('tienda', 6.102882391353436e-07), ('mano', 4.993267411107356e-07), ('dulce', 3.883652430861277e-07), ('galleta', 2.774037450615198e-

[1] `https://github.com/jocarema/Busqueda-Lexica`

07)

- *Tiempo.* ('policía', 4.993267411107356e-07), ('tienda', 4.3267278570012044e-07), ('gelatina', 4.161056175922797e-07), ('mamá', 3.606248685799758e-07), ('fresa', 2.342520513852834e-07)

- *Asociación.* ('gelatina', 7.212497371599516e-07), ('tienda', 6.102882391353436e-07), ('mano', 3.883652430861277e-07), ('dulce', 3.883652430861277e-07), ('galleta', 1.664422470369119e-07)

6.2.1. Centralidad Intermedia

Los resultados para el motor basado en Centralidad Intermedia[2] se muestran en la tabla 6-3.

Tabla 6-3: Resultados de la búsqueda léxica basado en un motor de CI y el grafo con pesos.

Función de peso	**p@1**	**p@3**	**p@5**
Tiempo(T)	0.3623	0.5507	0.6522
Frecuencia Inversa (FI)	0.6165	0.7419	0.7742
Fuerza de Asociación Inversa (FAI)	0.6558	0.8043	**0.8297**

En la tabla 6-4 se presentan los resultados de aplicar el motor de búsqueda basado en Centralidad Intermedia, pero tomando el grafo sin las funciones de peso sobre los lados del grafo[3]. Se puede observar que los resultados no son tan buenos como cuando se toma el grafo con sus respectivos pesos.

Tabla 6-4: Resultados de la búsqueda léxica basado en un motor de CI en el grafo sin peso.

p@1	**p@3**	**p@5**
0.3405	0.5621	**0.6648**

[2] https://github.com/jocarema/Busqueda-Lexica/blob/master/CI%20(FA%2CFAI%2CT).txt
[3] https://github.com/jocarema/Busqueda-Lexica/blob/master/CI.txt

6.2.2. Comunicabilidad de Centralidad Intermedia

La tabla 6-5 presenta los resultados obtenidos al utilizar el motor de búsqueda basado en CCI[4]. Este algoritmo trabaja con grafos sin peso en los lados.

Tabla 6-5: Resultados de la búsqueda léxica basado en un motor de CCI.

p@1	**p@3**	**p@5**
0.1532	0.3284	**0.4562**

Se nota una disminución de precisión respecto al algoritmo CI. Incluso, cuando en CI se trabaja con grafos sin peso.

6.2.3. Centralidad de Cercanía

Para esta implementación del motor de búsqueda se presentan tres tablas. La tabla 6-6 muestra los resultados utilizando la versión clásica de CC considerando las funciones de peso en los lados del grafo.

Tabla 6-6: Resultados de la búsqueda léxica basado en un motor de CC clásico y el grafo con pesos.

Función de peso	**p@1**	**p@3**	**p@5**
Tiempo(T)[5]	0.0802	0.2554	0.3576
Frecuencia Inversa (FI)[6]	0.2518	0.4635	0.5693
Fuerza de Asociación Inversa (FAI)[7]	0.24817	0.4890	**0.5985**

La tabla 6-7 presenta los resultados del motor de búsqueda basado en CC mejorado.

De acuerdo con los resultados anteriores, los mejores resultados se obtiene con la función de peso FAI y con la versión mejorada de CC. Se observa una ligera mejoría de la precisión

[4] https://github.com/jocarema/Busqueda-Lexica/blob/master/CCI.txt
[5] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Clasica(T).txt
[6] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Clasica%20(FI).txt
[7] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Clasica(FAI).txt
[8] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Mejorada(T).txt
[9] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Mejorada(FI).txt
[10] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Mejorada(FAI).txt

Tabla 6-7: Resultados de la búsqueda léxica basado en un motor de CC mejorado y el grafo con pesos.

Función de peso	**p@1**	**p@3**	**p@5**
Tiempo(T)[8]	0.0875	0.2554	0.3576
Frecuencia Inversa (FI)[9]	0.2700	0.4635	0.5729
Fuerza de Asociación Inversa (FAI[10]	0.2627	0.4927	**0.6058**

comparada con la versión clásica.

Finalmente, se presenta en la tabla 6-8 los resultados de las versiones clásicas y mejoradas, con el grafo sin pesos.

Tabla 6-8: Resultados de la búsqueda léxica basado en un motor de CC y un grafo sin pesos.

Tipo	**p@1**	**p@3**	**p@5**
Clásica[11]	0.1594	0.4311	0.5652
Mejorada[12]	0.1605	0.4452	**0.5839**

Se puede observar que la versión mejorada de CC es más precisa para la búsqueda léxica. Este comportamiento es consistente en todos los resultados anteriores de CC. Sin embargo, se puede observar mejores resultados cuando se trabaja con grafos pesados.

6.2.4. Centralidad Katz

La tabla 6-9 presenta los resultados del motor de búsqueda implementado con el algoritmo CK y con el grafo sin pesos [13].

Tabla 6-9: Resultados de la búsqueda léxica basado en un motor de CK.

p@1	**p@3**	**p@5**
0.1094	0.2189	**0.2810**

Los resultados obtenidos al utilizar los grafos con peso no fueron favorables para ningún

[11] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Clasica.txt
[12] https://github.com/jocarema/Busqueda-Lexica/blob/master/CC%20Mejorada.txt
[13] https://github.com/jocarema/Busqueda-Lexica/blob/master/CK.txt

caso de la búsqueda léxica con el motor basado en CK, es decir, las precisiones obtenidas en todos los casos fue de cero.

6.2.5. Centralidad de Vector Propio

Los resultados para el motor basado en CVP y con las tres funciones de peso se muestran en la tabla 6-10[14].

Tabla 6-10: Resultados de la búsqueda léxica basado en un motor de CPV y el grafo con pesos.

Función de peso	**p@1**	**p@3**	**p@5**
Tiempo(T)	0.1204	0.2956	**0.3941**
Frecuencia Inversa (FI)	0.1459	0.2810	0.3759
Fuerza de Asociación Inversa (FAI)	0.1313	0.2883	0.3759

En la tabla 6-11 se presentan los resultados de aplicar el motor de búsqueda basado en Centralidad de Vector Propio, pero tomando el grafo sin las funciones de peso sobre los lados del grafo[15].

Tabla 6-11: Resultados de la búsqueda léxica basado en un motor de CVP en el grafo sin peso.

p@1	p@3	p@5
0.1459	0.3138	**0.4197**

En esta implementación, los resultados del grafo sin peso fueron mejores en precisión que el mejor resultado usando el tiempo como función de peso del grafo.

6.2.6. Centralidad de Cercanía y Flujo Corriente

Los resultados para el motor basado en CCFC usando las tres funciones de peso[16] se muestran en la tabla 6-12.

[14] https://github.com/jocarema/Busqueda-Lexica/blob/master/CPV(FA%2CFAI%2CT).txt
[15] https://github.com/jocarema/Busqueda-Lexica/blob/master/CPV.txt
[16] https://github.com/jocarema/Busqueda-Lexica/blob/master/CCFC(FA%2CFAI%2CT).txt

Tabla 6-12: Resultados de la búsqueda léxica basado en un motor de CCFC y el grafo con pesos.

Función de pesado	**p@1**	**p@3**	**p@5**
Tiempo(T)	0.1350	0.2919	0.3832
Frecuencia Inversa (FI)	0.1350	0.2700	**0.3905**
Fuerza de Asociación Inversa (FAI)	0.1350	0.2737	**0.3905**

En la tabla 6-13 se presentan los resultados de aplicar el motor de búsqueda basado en CCFC, pero tomando el grafo sin las funciones de peso sobre los lados del grafo[17].

Tabla 6-13: Resultados de la búsqueda léxica basado en un motor de CCFC en el grafo sin peso.

p@1	**p@3**	**p@5**
0.1313	0.2846	**0.4051**

En esta implementación, los resultados del grafo sin peso fueron ligeramente mejores en precisión que el mejor resultado usando FI y FAI, como funciones de peso en el grafo.

6.2.7. Centralidad de Grado

La tabla 6-14 presenta los resultados obtenidos al utilizar el motor de búsqueda basado en CG[18]. Vale la pena recordar que este algoritmo no trabaja con grafos pesados.

Tabla 6-14: Resultados de la búsqueda léxica basado en un motor de CG.

p@1	**p@3**	**p@5**
0.1496	0.3029	**0.4416**

A pesar de la simplicidad del modelo basado en CG, se puede observar que arroja resultados alentadores.

[17] `https://github.com/jocarema/Busqueda-Lexica/blob/master/CCFC.txt`
[18] `https://github.com/jocarema/Busqueda-Lexica/blob/master/CG.txt`

6.3. Comparación con otros modelos de recuperación de información

Con la finalidad de evaluar la relevancia del modelo de búsqueda léxica, se realizaron experimentos con otros métodos conocidos de recuperación de información.

Primeramente, se comparó el desempeño del modelo con los resultado de un diccionario inverso. Para hacer esto, se usó el *OneLook Thesaurus*, que permite describir el concepto, y regresa una lista de palabras y frases relacionadas al concepto. Aunque existe una versión en español del recurso[19], es claramente superado por la versión en inglés, de tal forma que el uso de la versión en español ha sido descartado. Para usar *Onelook*, se tradujeron literalmente cada una de las definiciones del corpus, así como los conceptos objetivo, usando *Google Translator*. Las definiciones han sido manualmente probadas usando la aplicación web de *OneLook*[20].

Posteriormente, se compararon los resultados del modelo con aquellos obtenidos por el modelo base de recuperación de información usando búsqueda booleana. Los experimentos fueron realizados en dos corpus diferentes de referencia: a) Diccionario de la Real Academia Española RAE [2013], y b) Corpus de Normas de Asociación de Palabras para el Español de México [Arias-Trejo *et al.*, 2015].

El Motor de Recuperación Booleano[21] toma cada definición del corpus y genera una consulta juntando las palabras con conectores lógicos tipo *Y*, para obtener los documentos más relevantes que contengan todos los elementos en la búsqueda. Para este experimento, el motor primero busca un archivo conteniendo cada palabra de la definición. En caso de no encontrarla, la función *Quorum* [Cleverdon, 1984] es utilizada, es decir, otra búsqueda es realizada con todas las palabras menos una y cada combinación posible. El proceso continúa hasta encontrar una combinación que devuelva un documento.

Así como los modelos de búsqueda léxica del capítulo 5, la búsqueda booleana requiere varios corpus y tareas de pre-procesamiento en las definiciones. Sin embargo, es importante

[19] `http://www.rimar.io/`
[20] `https://www.onelook.com/thesaurus/`
[21] `https://github.com/jin-zhe/boolean-retrieval-engine`

mencionar que una condición de paro se estableció en el ciclo de búsqueda, refiriendo que las consultas deben tener un mínimo de dos palabras en la definición, porque devolvería muchos conceptos que podrían coincidir con cualquier palabra.

Se realizaron experimentos adicionales con uno de los algoritmos de recuperación de información más exitosos, *Okapi BM25*, como se mencionó anteriormente, está basado en modelos probabilísticos y desarrollado en los setentas por Robertson y Sparck Jones [1976]. El algoritmo implementado siguiendo Robertson y Zaragoza [2009] está basado en el método de bolsa de palabras. Dada una consulta, devuelve una lista ordenada de documentos de acuerdo a su relevancia para la tal consulta. Se aplicó considerando como un documento cada conjunto de respuestas de un estímulo.

Finalmente, se realizó un experimento usando vectores pre-entrenados. Se tomó como base la primera etapa del trabajo *Computing Associative Responses* (CAS) [Ghosh *et al.*, 2014]. Este trabajo involucra la generación de una lista ordenada de respuestas a un conjunto de palabras estímulo. En este caso, los estímulos fueron las palabras en una definición y la respuesta inferida es el concepto que se trata de encontrar en el diccionario onomasiológico. Los vectores de la implementación de CAS fueron en español usando *fastText* [Bojanowski *et al.*, 2017][22]. *FastText* es un método que ha sido diseñado para mejorar el desempeño de *word2vec*, basado en el modelo *skip-gram*, con la diferencia que en esta aproximación cada palabra es representada como una bolsa de n-gramas de caracteres. Mikolov *et al.* [2018] expone que los modelos entrenados con *fastText* exhiben el mejor grado de exactitud comparado con otros sistemas. Para este experimento se usó la representación vectorial de cada palabra en la definición y se calculó la similitud de un concepto objetivo mediante la medición de la similitud coseno entre las palabras de la definición y aquellas en el recurso de *fastText*. Formalmente se describe de la manera siguiente:

Sea $S = \{x_i, \ldots, x_j\}$ las palabras de la definición

$$sim(r, S) = \frac{1}{|S|} \times \sum_{i=1}^{|S|} \frac{x_i \cdot r}{|x_i| \cdot |r|} \tag{6-1}$$

[22]https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

Es importante mencionar que las definiciones fueron probadas sin palabras funcionales y en su forma lematizada. Con este experimento únicamente se obtuvieron 88 conceptos objetivo de un total de 280 definiciones, en una posición promedio de 166.

Tabla 6-15: Resultados comparativos de precisión.

Método	**P@1**	**P@3**	**P@5**
OneLook	0.1151	0.2050	0.2554
CI (FAI)	**0.6558**	**0.8043**	**0.8297**
NAP Booleano	0.3776	0.3776	0.3776
RAE Booleano	0.0359	0.0359	0.0359
BM25	0.3300	0.4900	0.5700
CAS	0	0.0070	0.0140
Wantwords	0.1090	0.1709	0.2363

6.3.1. Discusión

Los resultados alcanzados usando los cinco métodos bases mencionados: OneLook, modelos booleanos de recuperación de información, BM25, el sistema de dos etapas con vectores pre-entreenados y Wantwords, son reportados en la tabla 6-15, donde son comparados con el mejor resultado obtenido por el modelo de búsqueda léxica.

Con la idea de realizar una búsqueda booleana sobre el corpus NAP, cada estímulo fue considerado como un documento y sus respuestas fueron el contenido de los documentos. Para buscar sobre la RAE, cada definición en el diccionario fue considerada como un documento. La búsqueda léxica mostró un mejor desempeño que el diccionario inverso *OneLook* cuando la búsqueda es realizada sobre el corpus NAP. Sin embargo, cuando la búsqueda es realizada sobre la RAE, los resultados son muy bajos. Se considera que este comportamiento es debido a la corta naturaleza del vocabulario técnico de un definición de diccionario.

Respecto a BM25, el algoritmo muestra un desempeño bastante bueno, sin embargo, es lejano al Modelo de Búsqueda Léxica. Por el contrario, CAS, implementado con los vectores de *fastText*, tiene los peores resultados. A pesar de que los vectores de palabras son muy útiles

en otras tareas relacionadas con semántica, el modelo falla en la búsqueda léxica. Esto puede ser causado, entre otras causas, por la gran diversidad de vocabulario y la presencia de todas las formas flexionadas de una palabra en este tipo de recursos. No se realizó la segunda etapa del experimento con este método (CAS), porque únicamente realiza una ordenación de los resultados obtenidos en la primera etapa con la finalidad de tener las palabras objetivo en las primeras posiciones. En este procedimiento, en el mejor de los escenarios, la mejora del desempeño podría alcanzar un 30 % y no sería significativo para vencer el Modelo basado en CI.

En cuanto a los resultados obtenidos con *WantWords*, se puede observar que son similares a los obtenidos con *Outlook*. La forma de evaluación fue similar en ambos casos, es decir, traduciendo directamente las definiciones del corpus al idioma inglés y probándolas en sus respectivas aplicaciones. El motor interno de *WantWords* [Qi *et al.*, 2020] está basado en *BERT (Bidirectional Encoder Representations from Transformers)* [Devlin *et al.*, 2018], el cual es uno de los modelos más populares en PLN por su efectividad. Sin embargo, en la tarea del diccionario inverso, existen mejores aproximaciones que superan sus resultados, al menos en lo concerniente a las pruebas realizadas en este trabajo de investigación. En parte, los resultados tan bajos pueden deberse a que las definiciones dadas por los usuarios tienen algunas veces connotaciones culturales que hacen las descripciones muy apegadas al español de México. Es así, que traducirlas y aplicarlas a un diccionarios en inglés propicia la pérdida de precisión en sus resultados.

Las tablas 6-16 y 6-17 muestran lo que cada sistema devolvió con dos definiciones diferentes de las palabras *queso* y *abeja*. Los resultados indican que el diccionario inverso *OneLook* no es adecuado para resolver este problema, devolviendo el concepto correcto en los primeros 5 resultados solamente el 25 % de las veces. Además, revisando cuidadosamente los resultados de cada uno de los sistemas, una limitación del Modelo de Búsqueda Léxica puede ser observada. Los resultados son los mejores cuando se usa el modelo evaluación $p@k$; sin embargo, cada método que trabaja sobre NAP tiene algunos resultados no consistentes. Por ejemplo, cuando se comparan los resultados de las pruebas de BM25 para la palabra *queso* con NAP y RAE, el

modelo devuelve algunas palabras que son difíciles de explicar con NAP: *pelo*, *colores*, mientras que con RAE solamente muestra resultados no coherentes, *champús*. A pesar que el sistema es rápido, eficiente y demuestra un alto desempeño, la estructura del recurso usado favorece el hecho de que dos palabras que no están realmente relacionadas por asociación tienen un camino corto entre ellas porque comparten una palabra conectada. Esto se espera que sea una limitación del Modelo de Búsqueda Léxica, que puede ser minimizada mediante la realización de algún tipo de filtro léxico en el futuro.

En la tabla 6-18 se presenta la comparación de todos los motores de búsqueda que se implementaron basados en los algoritmos de centralidad de grafos, tomando los mejores resultados en cada caso.

El mejor resultado es obtenido basado en el motor de Centralidad Intermedia. Es claro que cuando el modelo busca sobre los grafos pesados con FI y FAI, los resultados son más altos que cuando se busca sobre el grafo pesado con T. Además, la búsqueda con el grafo FAI logra la precisión mas alta de todas las medidas de evaluación. Este era un resultado esperado. De acuerdo con la psicolingüística, el tiempo de reacción no necesariamente indica relacionalidad entre estímulos y respuestas, aunque la intuición indique que pueda haber alguna conexión. Sin embargo, hasta ahora, ningún estudio ha podido establecer una relación sistemática.

Psicólogos concuerdan que la fuerza de asociación es la medida que implica una relación cognitiva entre dos términos, y esta idea está reflejada en los resultados del modelo. La frecuencia está relacionada cercanamente a la FA, pero adolece de generalización de esta última función.

Respecto a los algoritmos de centralidad, se tiene el siguiente orden (basado en p@5) donde el primer lugar corresponde al mejor modelo, terminando la lista con la centralidad menos precisa en la tarea del diccionario inverso.

1. Centralidad intermedia
2. Centralidad de cercanía
3. Comunicabilidad de centralidad intermedia
4. Centralidad de grado

Tabla 6-16: Resultados para *queso*.

	Queso	
Definición **Método**	Alimento elaborado con leche. Existen diferentes tipos: manchego, cotija, panela entre otros.	El producto que se saca de la leche de la vaca
Modelo CI (FI)	1. queso 2. torta 3. colores 4. pelo 5. dulce	1. queso 2. torta 3. calabaza 4.colores 5.pelo
Modelo CI (T)	1. queso 2. colores 3. vaca 4. comer 5. blanco	1. queso 2. calabaza 3. colores 4. alimento 5. vaca
Modelo CI (FAI)	1. queso 2. pelo 3. ratón 4. pastel 5. colores	1. queso 2. calabaza 3. leche 4. pelo 5. mercado
BM25 NAP	1. queso 2. torta 3. pelo 4. colores 5. mamila	1. a leche de la vaca 2. tienda 3. calabaza 4. queso 5. pala
BM25 RAE	1. queso 2. chéster 3. gorgonzola 4. brandy 5. champús	1. mantequilla 2. cuajada 3. lacteado, da 4. lácteo, a 5. natilla
OneLook	1. atom 2. expenses 3. aphid 4. rounds 5. meal	1. strip 2. ghee 3. buttermilk 4. stroking 5. mess
CAS	1. piloncillo 2. nixtamalizado 3. nixtamalización 4. saborización 5. quesillo	1. leche 2. producto 3. pasteurizar 4. trío 5. sacar
NAP Booleano	Consulta: alimento **Y** leche **Y** manchego **Y** panela	Consulta: leche **Y** vaca
WantWords	1. salsa 2. tostada 3. gastronomy 4. manioc 5. croquette	1. butterfat 2. ghee 3. kefir 4. roughage 5. soymilk

5. Centralidad de vector propio

6. Centralidad de cercanía y flujo corriente

7. Centralidad Katz

La centralidad intermedia obtiene el primer lugar. El hecho de que la versión de este algorit-

Tabla 6-17: Resultados para *abeja*.

	Abeja	
Definición **Método**	Insecto volador rayado que produce miel	Insecto volador amarillo y negro
Modelo CI (FI)	1. cuchara 2. circo 3. luz 4. grande 5. feo	1. cuchara 2. circo 3. palo 4. martillo 5. manzana
Modelo CI (T)	1. abeja 2. mariposa 3. ardilla 4. cacahuate 5.conocimiento	1. abeja 2. martillo 3. ardilla 4. agua 5. cacahuate
Modelo CI (FAI)	1. abeja 2. mariposa 3. conocimiento 4. minifalda 5. 2.0	1. abeja 2. mariposa 3. arańa 4. tractor 5. plastilina
BM25 NAP	1. palomita 2. crayola 3. circo 4. cebra 5. pluma	1. mariposa 2. helicóptero 3. ardilla 4. palomita 5. circo
BM25 RAE	1. fatula 2. lapizar 3. eraje 4. meloja 5. bresca	1. mapanare 2. fatula 3. doral 4. agüío 5. cacuy
OneLook	1. bee 2. manna 3. moth 4. virgin 5. dor	1. wasp 2. gnat 3. bee 4. whippoorwill 5. slug
CAS	1. rayar 2. insecto 3. rayadura 4. miel 5. espesarla	1 amarillo 2. negro 3. amarillo/naranja 4. amarillo/blanco 5. anaranjado
NAP Booleano	Consulta: insecto **Y** rayar **Y** miel	Ninguna combinación de términos devolvió abeja
WantWords	1. ackfly 2. hawkmoth 3. firefly 4. ladybug 5.bumblebee	1. damselfly 2. damselfly 3. firefly 4. dragonfly 5. housefly

mo esté basada en subconjuntos, hace que la búsqueda quede acotada y mejore la precisión del diccionario inverso. En segundo lugar, se tiene la centralidad de cercanía, en su descripción se afirma que la medida de centralidad considera que un nodo es más importante si puede alcanzar todos los otros nodos en el menor número de pasos; la reducción del grafo, que se basa en la definición dada por un usuario, permite identificar los nodos más relevantes tomando en cuenta

Tabla 6-18: Resultados de la búsqueda léxica basado en algoritmos de centralidad.

Algoritmo	**p@1**	**p@3**	**p@5**
CI (FAI)	0.6558	0.8043	**0.8297**
CCI (sin peso)	0.1532	0.3284	0.4562
CC - Mejorado (FAI)	0.2627	0.4927	0.6058
CK (sin peso)	0.1094	0.2189	0.2810
CVP (sin peso)	0.1459	0.3138	0.4197
CCFC (sin peso)	0.1313	0.2846	0.4051
CG (sin peso)	0.1496	0.3029	0.4416

todas la conexiones y no solamente las inmediatas. Con base en las pruebas realizadas, se puede ver que las dos primeras medidas de la lista son mejores que los modelos de recuperación tradicionales, incluso CC vence el BM-35 que es el modelo clásico más competitivo. Es importante indicar que el modelo basado en CC tomado en todas sus versiones: clásico, mejorado, con pesos y sin ellos, supera los modelos clásicos (no basados en grafos) contra los que se comparó.

El tercer lugar corresponde a la comunicabilidad de centralidad intermedia que no solamente toma en cuenta los caminos más cortos, considera todos aquellos que pasen por un nodo pero dándoles un peso dependiendo de su longitud. Es importante mencionar que este algoritmo no trabaja con los pesos de los lados del grafo, por lo que en los casos que tenga una norma con ausencia de medidas que asocien los estímulos con sus respuestas, es una buena opción para tomar en cuenta en el modelo de búsqueda léxica. El siguiente lugar corresponde a la centralidad de grado, sorprendentemente a pesar de su simpleza, obtiene el 4o lugar del listado, significando que las palabras proporcionadas por un usuario durante la búsqueda, logran varias conexiones directas a la palabra objetivo. El quinto lugar es para la centralidad de vector propio, en su descripción se establece que es una medida similar a la Centralidad Katz (8.° lugar). Es coincidente en los resultados que fueron poco favorables para ambas aproximaciones. La CVP ordena los nodos con base en aquellos que están cerca de nodos influyentes, lo cual no resulta significativo para encontrar palabras objetivo dentro del modelo. Finalmente, el 7.° lugar de la centralidad de cercanía y flujo corriente, hace una combinación de las medidas de

centralidad de intermedia y la de cercanía, midiendo el flujo de la información a través de la red y emulando algunos fenómenos como si fuera una red eléctrica, que según los resultados, no provee de resultados precisos para la tarea del diccionario inverso.

6.3.2. Evaluación del modelo en otro idioma

De acuerdo con los resultados anteriores, se puede apreciar que la mejor versión de los modelos basados en grafos es la versión lograda con el modelo basado en centralidad intermedia. Por esta razón, al hacer las evaluaciones en inglés, únicamente se implementó el modelo basado en CI.

El modelo es el mismo que ha sido descrito en la sección 5.2.1, lo único que cambia es la lista de palabras vacías, ahora para inglés, así como el lematizador. En cuestión del corpus de Normas de Asociación, se usaron dos: el de *Edinburgh Associative Thesaurus (EAT)* y el de *South Florida Free Association Norms (USF)*, cada uno descrito en las secciones 2.5.2.1 y 2.5.2.2 respectivamente.

Los resultados son mostrados en la tabla 6-19 y la tabla 6-20. De manera general, cuando el modelo realiza búsquedas con pesos basados en fuerza de asociación inversa, los resultados son más altos en ambos corpus, esto es consistente con los resultados obtenidos en español. Recordemos que la frecuencia está relacionada con la fuerza de asociación, pero adolece de generalización. Respecto a las normas de asociación, los mejores resultados se alcanzaron usando las normas *USF*.

Tabla 6-19: Resultados de búsqueda léxica basados en CI y normas EAT

Función de peso	**p@1**	**p@3**	**p@5**	**p@10**
Frecuencia Inversa (FI)	**0.152**	0.186	0.220	0.237
Fuerza de Asociación Inversa (FAI)	**0.152**	**0.220**	**0.237**	**0.254**

Al igual que se hizo con el corpus en español, se evaluó la relevancia del método comparando con otros modelos de recuperación de información, como se puede apreciar en la tabla 6-21.

El algoritmo BM25 mostró mejores resultados comparado con el diccionario inverso *One-*

Tabla 6-20: Resultados de búsqueda léxica basados en CI y normas USF

Función de peso	**p@1**	**p@3**	**p@5**	**p@10**
Frecuencia Inversa (FI)	0.236	0.309	**0.418**	0.436
Fuerza de Asociación Inversa (FAI)	**0.290**	**0.363**	**0.418**	**0.5272**

Tabla 6-21: Resultados comparativos con otros modelos para inglés

Método	**P@1**	**P@3**	**P@5**	**P@10**
OneLook	0.202	0.347	0.376	0.434
Modelo CI corpus USF y peso FAI	**0.290**	0.363	0.418	**0.5272**
BM25 con EAT	0.257	0.357	0.414	0.471
BM25 con USF	0.257	**0.400**	**0.457**	0.514
Wantwords	0.057	0.200	0.257	0.357

Look cuando la búsqueda es realizada sobre los corpus de normas de asociación. El BM25 fue implementado usando tanto EAT como USF. Al igual que en español, para cada estímulo se construyó un documento contiendo todas las respuestas del recurso psicolingüístico. Los resultados más altos son consistentes con los observados en el modelo de búsqueda léxica, las normas USF mostraron los mejores desempeños con este algoritmo de recuperación de información. Se puede apreciar que este algoritmo es el más competitivo comparado con el modelo basado en CI, sin embargo, se superaron los resultados en p@1 y p@10, por el contrario, hubo un decremento de precisión en p@3 y p@5. Finalmente, *Wantwords* arrojó resultados incluso por debajo de *OneLook*, este mismo comportamiento se pudo observar en el desempeño obtenido para el idioma español.

6.4. Vectorización de normas de asociación

Los vectores de palabras o *embeddings* son herramientas poderosas en muchas tareas de PLN. Tomando como base los corpus de normas de asociación de palabras, es posible aprender vectores de palabras, usando el grafo y el algoritmo *node2vec*. A pesar de que las normas de asociación de palabras construidas con metodologías clásicas son recursos complejos de recolectar por

cuestiones de tiempo y el desarrollo del proceso completo para compilarlas, el entrenamiento de los vectores es un proceso rápido y eficiente. La evaluación de los vectores se realizó mediante dos maneras: intrínseca y extrínseca. La evaluación intrínseca se realizó con unos conjuntos de datos que miden similitud entre palabras. En cuanto a la evaluación extrínseca, fue realizada midiendo la calidad de los vectores aplicados a tareas de tranferencia: análisis de sentimientos, detección de paráfrasis, etc.

6.4.1. *Node2vec*

El algoritmo *node2vec* [Grover y Leskovec, 2016] encuentra un mapeo $f : V \rightarrow \mathbb{R}^d$ tranformando los nodos de un grafo en vectores de d-dimensiones. Define un vecindario $N_s(u) \subset V$ para cada nodo $u \in V$ a través de una técnica de muestreo S. El objetivo del algoritmo es maximizar la probabilidad de observar nodos consecutivos en un camino aleatorio de longitud fija.

La estrategia de muestreo diseñada en *node2vec* permite explorar vecindarios con caminos aleatorios. Parámetros como p y q permiten controlar el cambio entre búsquedas *primero-en-anchura* y *primero-en-profundidad* en el grafo. El parámetro p controla la probabilidad de revisitar un nodo en el camino y el parámetro q permite a la búsqueda diferenciar entre nodos entrantes o salientes. Los valores que se usaron en el trabajo fueron $p = 1$ y $q = 1$. Se usó la implementación disponible en el sitio web[23] del proyecto de *node2vec*, con los valores por defecto para todos los parámetros. Se examinó la calidad de los vectores con diferentes dimensiones d y diferentes longitudes de camino l.

6.4.2. Wan2Vec

Wan2Vec [Bel-Enguix *et al.*, 2019] es el acrónimo que se utilizó para identificar el proceso de vectorización de normas de asociación: *Wan* significa *Word Association Norms*, 2 para significar el "to" y *Vec* para *vectors. Word Association Norms to Vectors.*

En esta sección se presenta la construcción de vectores, en el idioma inglés, para ello se

[23] `snap.stanford.edu/node2vec/`

utilizaron las normas EAT descritas en la sección 2.5.2.1 del capítulo 2. A pesar que el corpus podría ser considerado algo antiguo, ya que data de los 70's, es un recurso bastante completo y balanceado. Se usó la versión XML del recurso sociolingüístico[24]. Los *embeddings* de palabras entrenados con el *EAT* están disponibles el página de Github[25].

La representación del grafo basado en el corpus *EAT* se define formalmente como:
$G = \{V, E, \phi\}$ donde:

- $V = \{v_i | i = 1, ..., n\}$ es un conjunto finito de nodos de longitud n, $V \neq \emptyset$, correspondientes a los estímulos y sus correspondientes respuestas.
- $E = \{(v_i, v_j) | v_i, v_j \in V, 1 \leq i, j \leq n\}$, es el conjunto de lados.
- $\phi : E \rightarrow \mathbb{R}$, es la función de peso sobre los lados.

El grafo creado es no dirigido, por lo que cada estímulo está conectado a sus respuestas sin ningún orden establecido.

Se usaron dos funciones diferentes (ϕ) para asignar peso a los lados: frecuencia y fuerza de asociación [Nelson *et al.*, 1998]. Solamente la primera está incluida en los datos proporcionados por EAT. El último fue calculado de los resultados de la frecuencia para cada respuesta asociada a su estímulo.

Frecuencia (ϕ_F) Cuenta el número de ocurrencias de cada respuesta asociada a su estímulo en todo el corpus. Con la finalidad de usar esta información, se calculó la frecuencia inversa (ϕ_{FI}), que se define de la siguiente manera: sea ϕ_F la frecuencia de una palabra respuesta, y $\Sigma\phi_F$ la suma de las frecuencias de todas las palabras conectadas al mismo estímulo $\phi_{FI} = \Sigma\phi_F - \phi_F$.

Fuerza de asociación (ϕ_{FA}) Establece la relación entre la frecuencia (ϕ_F) y el número de respuestas para cada estímulo. Se calcula de la siguiente manera: sea ϕ_F la frecuencia de una palabra respuesta y $\Sigma\phi_F$ la suma de las frecuencias de las respuestas al mismo

[24] `http://www-rali.iro.umontreal.ca/rali/?q=en/XML-EAT`
[25] `https://github.com/jocarema/Wan2Vec`

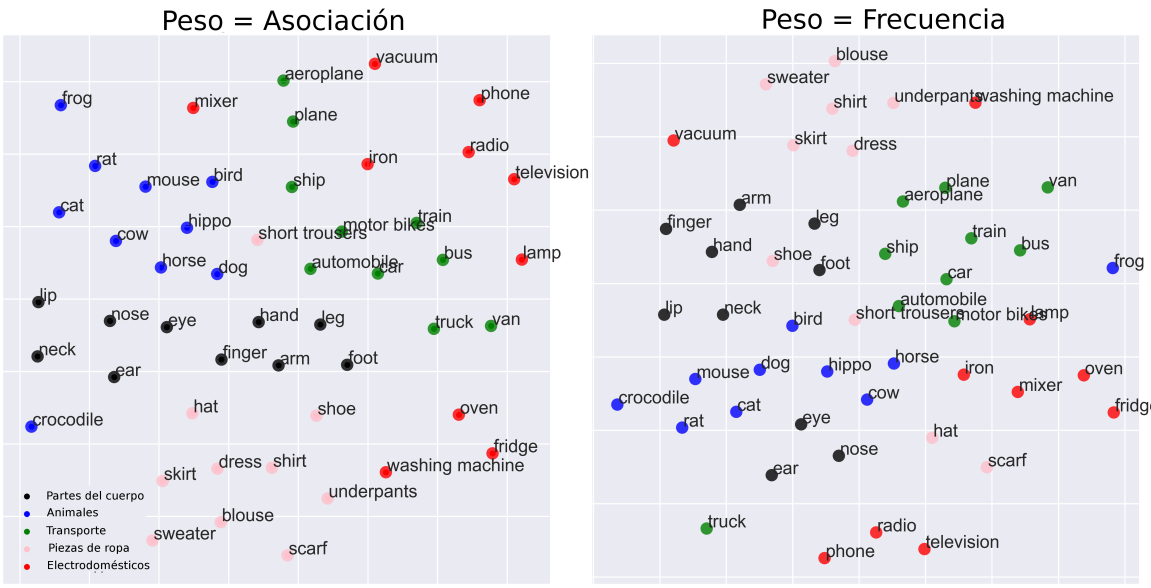

Figura 6-1: Proyección vectorial de palabras para 5 grupos semánticos (10 de cada uno). La codificación de colores es: animales - rojo, transportes - negro, partes del cuerpo - azul, electrodomésticos - verde, piezas de ropa - rosa.

estímulo (el total de respuestas), la fuerza de asociación (ϕ_{FA}) de la palabra para tal estímulo es dada por la fórmula:

$$\phi_{FA} = \frac{\phi_F}{\Sigma\phi_F}$$

Para los experimentos, se usó la fuerza de asociación inversa, ϕ_{FAI} con la finalidad de preparar el sistema para trabajar con los algoritmos basado en grafos:

$$\phi_{FAI} = 1 - \frac{\phi_F}{\Sigma\phi_F}$$

Es importante señalar que para la creación de vectores basados en grafos se usaron ϕ_{FI} y ϕ_{FAI} porque el algoritmo *node2vec* recorre los caminos más cortos.

6.4.2.1. Evaluación intrínseca

Las evaluaciones intrínsecas evalúan la habilidad de los vectores de palabras de capturar las relaciones sintácticas o semánticas de las palabras [Baroni *et al.*, 2014]. La hipótesis de la

evaluación intrínseca establece que palabras similares deben tener representaciones similares. Por lo tanto, para evaluar la similitud primero se realizó una visualización de una muestra de los vectores de palabras, usando una proyección *t-distributed Stochastic Neighbor Embedding (T-SNE)*[26] en un espacio de dos dimensiones. El experimento fue realizado usando ϕ_F y ϕ_{FA} como funciones de peso, nótese que para los experimentos de visualización no se usó la versión inversa de las funciones de peso. Se examinaron 50 palabras, divididas en cinco grupos semánticos. La figura 6-1 muestra cómo las palabras relacionadas están agrupadas. Cuando se usó la función ϕ_{FA} como peso del grafo, el método en su mayoría reúne los elementos de los 5 grupos semánticos. Se puede observar que los animales y partes del cuerpo están agrupados juntos. Las piezas de ropa tienen su propio espacio, excepto para *short trousers* que aparece aislado. Un fenómeno interesante es que *washing machine* está cercana a piezas de ropa y realmente podría pertenecer a dos grupos semánticos: electrodomésticos y ropa.

Cuando se usa F como función de peso, los bordes de los grupos no están tan bien definidos. *Washing machine* aun tiene una localización cercana a algunas piezas de ropa, pero esta categoría esta dispersada sin ningún orden en el espacio bidimensional. *Truck* y *frog* están también lejanos de su grupo esperado.

Similitud y relacionalidad Adicionalmente, se evaluó la habilidad de *Wan2vec* de capturar relaciones semánticas a través de tareas de similitud. Se usaron los datos de *WordSim-353* [Finkelstein *et al.*, 2001] como punto de referencia, compuesto por 353 pares de términos relacionados con puntuaciones de similitud dado por humanos. La lista no distingue entre los conceptos de similitud y relacionalidad. Agirre *et al.* [2009] dividieron la lista en dos conjuntos diferentes, uno con las puntuaciones de relacionalidad, conteniendo 252 pares [EN-WS-353-REL], y otro con 203 pares ligados por similitud [EN-WS-353-SIM]. Existe un traslape entre las listas, todos los pares de términos pertenecen a los 353 pares de *WordSim 353*.

WordSim-353 está basado en trabajos de los 90's, fue elaborado en USA. Debido al tiempo y diferencias geográficas, algunas palabras de esta lista no están incluidas en el EAT, una

[26] `http://scikit-learn.org/stable/modules/manifold.html#t-sne`

colección británica de cerca de los 70's. Este hecho no es debido a la falta de expresividad en la norma, pero sí es causado porque algunos objetos, personas e ideas no existían cuando EAT fue recolectado, o fueron utilizados en contextos diferentes. Un ejemplo puede ser *hardware*, un neologismo en los 90's, o *Maradona*, la estrella de fútbol de los 80's. Como resultado, 183 pares de la lista de similitud están en el corpus EAT, mientras que 214 de los 253 están en los datos de relacionalidad. Para tratar con la ausencia de las palabras del conjunto de prueba en el EAT, se introdujo el concepto de traslape en los experimentos y se calculó el número total de palabras comunes entre las listas que están siendo comparadas. Las otras se excluyeron de la evaluación. En principio, tener grandes valores de traslape es una característica positiva para *Wan2vec*.

También se probó el método con SimLex-999 [SimLex-999] [Hill *et al.*, 2015], un recurso que contiene 999 pares de palabras y explícitamente cuantifica similitud de una forma tal que los pares relacionados por asociación o relacionalidad tiene un valor bajo. El traslape entre EAT y SimLex-999 es 968, lo que significa que casi cada palabra en la prueba está cubierta por *Wan2vec*.

El resto de los datos de puntos de referencia miden la relacionalidad de palabras, más que la similitud.

Los datos de *Amazon Mechanical Turk* [MTurk-287] [Radinsky *et al.*, 2011] consisten de 287 pares de palabras, elaborados en colaboración con trabajadores turcos de Amazon. Este modelo de relacionalidad captura información semántica basada en episodios de tiempo. El traslape con EAT es de 203.

MEN [MEN-TR-3k] [Bruni *et al.*, 2012] es un parámetro de referencia que consiste en 3,000 pares de palabras seleccionadas al azar. Cada par está calificado en una escala normalizada de [0,1], a través de calificaciones obtenidas mediante colaboración abierta en *Amazon Mechanical Turk*. En esta caso, se obtuvo un alto traslape, 2,727 de 3,000 pares de palabras.

El conjunto MTURK-771 [MTurk-771] [Halawi *et al.*, 2012] evalúa la relacionalidad entre 771 pares de palabras. Estos datos fueron obtenidos con la evaluación de trabajadores turcos mecánicos de Amazon. *EAT* cubre 698 pares de palabras de este conjunto de datos.

El RG65 [RG-65][Rubenstein y Goodenough, 1965] recolecta datos para 65 pares de palabras no técnicas. Su objetivo fue evaluar los juicios y percepciones sobre sinonimia, y debido a esto se usaron diferentes pares de un rango alto de sinonimia hasta palabras totalmente no relacionadas. Cada par de palabras en RG65 también está en el EAT.

Datos	***n***	***Traslape***	**Dimensiones de vectores Wan2vec**						
			25	50	100	128	200	300	1000
MC-30	30	30	0.663	0.824	0.781	0.811	0.781	**0.850**	0.757
WS-353-SIM	204	183	0.682	0.763	**0.783**	0.748	0.767	0.744	0.727
MTurk-287	287	203	**0.727**	0.708	0.698	0.683	0.670	0.647	0.562
WS-353-REL	253	214	0.598	**0.695**	0.692	0.666	0.685	0.645	0.581
MEN-TR-3k	3000	2720	0.783	0.800	**0.803**	0.794	0.791	0.762	0.708
SimLex-999	999	968	0.429	0.454	0.495	0.507	**0.519**	0.5127	0.505
MTurk-771	771	698	0.679	0.708	**0.731**	0.717	0.722	0.700	0.667
RG-65	65	65	0.706	**0.816**	0.768	0.805	0.760	0.797	0.710
Promedio			0.659	**0.721**	0.719	0.716	0.712	0.707	0.652

Tabla 6-22: Correlación de Spearman entre las predicciones de *Wan2vec* (basados en similitud coseno) y los conjuntos de referencia. El grafo fue construido usando ϕ_{FI} como función de peso.

Finalmente, se evaluaron los vectores de *Wan2Vec* con el MC-30[MC-30] [Miller y Charlees, 1991]. Esta lista contiene 30 pares de palabras, todas ellas incluidas en *EAT*.

La tabla 6-22 reporta la correlación de Spearman entre los conjuntos de comparación descritos anteriormente y los vectores de palabras de diferentes dimensiones aprendidos en grafos no dirigidos del EAT, siendo ϕ_{FI} el peso de los lados. La tabla 6-23 presenta el desempeño basado también en coseno de los vectores *Wan2vec* usando ϕ_{FAI} como función de peso. Para cada tabla, se proporciona el número total de pares (n) y el número n de traslape.

En estas tablas se puede apreciar que, en general, los vectores de *Wan2vec* entrenados en el grafo con peso ϕ_{FAI} obtienen mejor correlación que los vectores entrenados en el grafo con peso ϕ_{FI}. Esto era algo esperado porque, generalmente, la fuerza de asociación es una normalización de la frecuencia que debe dar una idea más general en la relación entre las dos palabras. Sin embargo, y por la misma razón, los resultados no son significativamente diferentes entre las dos

Datos	***n***	***Traslape***	**Dimensiones de vectores *wan2Vec***						
			25	50	100	128	200	300	1000
MC-30	30	30	0.649	0.772	0.792	0.820	0.848	**0.858**	0.740
WS-353-SIM	204	183	0.695	0.751	0.777	0.763	**0.777**	0.762	0.684
MTurk-287	287	203	**0.727**	0.716	0.697	0.692	0.661	0.666	0.569
WS-353-REL	253	214	0.622	0.696	**0.705**	0.685	0.698	0.658	0.571
MEN-TR-3k	300	272	0.782	0.803	**0.807**	0.799	0.785	0.764	0.714
SimLex-999	999	968	0.420	0.461	0.490	0.500	0.510	**0.510**	0.503
MTurk-771	771	698	0.676	0.712	0.730	**0.736**	0.719	0.697	0.663
RG-65	65	65	0.750	**0.821**	0.798	0.785	0.788	0.791	0.715
Promedio			0.665	0.716	**0.724**	0.722	0.723	0.713	0.645

Tabla 6-23: Correlación de Spearman entre las predicciones de *Wan2vec* (basados en similitud coseno) y los conjuntos de referencia. El grafo fue construido usando ϕ_{FAI} como función de peso.

funciones de peso. Resumiendo, ϕ_{FAI} puede ser considerado la medida de peso en los lados del grafo por defecto, aunque en una aproximación simple ϕ_{FI} funcionaría también sin una gran pérdida de expresividad.

Como se mencionó anteriormente, se probaron únicamente las palabras en la intersección entre las listas de prueba y las normas de EAT, la columna n en las tablas 6-22 y 6-23 corresponden al número de pares en los conjuntos de datos, mientras que la tercera columna, *traslape*, muestra el número de pares que también pueden ser encontrados en EAT. Asimismo, se puede apreciar que existe un traslape significativo entre los pares de palabras en cada conjunto de datos y el vocabulario de los vectores de *Wan2vec*. Esto significa que, en general, a pesar de tener un tamaño reducido en el recurso (solamente 20,445 nodos) el modelo ha tenido un alto nivel de expresividad, Sobre las dimensiones de los vectores, 50 y 100 mostraron ser los más eficientes. Con ϕ_{FI}, los vectores de dimensión 50 y 1000 tienen mejor desempeño que con ϕ_{FAI}. Sin embargo, esta última función de pesado alcanza mejor resultado que con el resto. El valor por defecto de *node2vec* está establecido en 128, pero para *Wan2Vec* los vectores más pequeños parece que tienen un mejor desempeño. No está claro si esto puede ser causado por el

relativamente pequeño número de nodos en el grafo. Cuando se usa ϕ_{FI} como función de peso, los vectores de 50 dimensiones tienen un promedio más alto en la correlación de Spearman, aunque los vectores con dimensión 100 obtuvieron los mejores resultados en 3 de los corpus. Por otro lado, los vectores con 100 dimensiones de *Wan2vec* parecen ser suficientes en promedio y en número de mejores resultados. Sin embargo, los mejores resultados del experimento son los alcanzados con los vectores de dimensión 300 con los datos de MC-30.

Desde el punto de vista de los parámetros de comparación, las similitudes obtenidas con *Wan2vec* alcanzaron correlaciones por encima de 0.8 con MC-30, RG-65, y MEN-TR-3k usando ϕ_{FI} y ϕ_{FAI} como funciones de peso. Esto es una sorpresa ya que MC-30 y RG-65 son los conjuntos más pequeños, mientras que MEN-TR-3k es el más grande. En principio, se esperaba tener un mejor desempeño con los conjuntos de prueba más pequeños. Probablemente, las características de los diferentes grupos de prueba, incluyendo el año de su elaboración, los rasgos sociodemográficos de los participantes y la orientación temática de las palabras tiene un impacto en los resultados.

Por otro lado, los peores resultados son alcanzados con el recurso SimLex-999. Esto no de debido al traslape reducido (es uno de los mejores, 96 %). Dado que el objetivo del conjunto de prueba SimLex es medir únicamente similitud, las palabras psicológicamente relacionadas tienen puntuaciones más bajas. Se cree que esta particularidad afectó el rendimiento de este recurso, ya que en EAT las palabras están supuestamente asociadas por relacionalidad. Sin embargo, esta idea no es consistente con los resultados obtenidos con WS-353-SIM y WS-353-REL, donde el primero ha superado el segundo en cada experimento.

En general, las puntuaciones más altas son alcanzadas con el MC-30. El hecho de que cada palabra en la lista se encuentre también en el EAT no está relacionado con el resultado porque también es el mejor cuando los pares faltantes son removidos. En el futuro, se deben realizar más investigaciones sobre estos resultados para conocer la dinámica de las asociaciones.

Se realizó un experimento adicional para medir el rendimiento y la adaptación de *Wan2vec* en diferentes algoritmos de similitud. Se reprodujeron las mismas pruebas que se describieron anteriormente; pero, ahora, en lugar de medir la correlación de dos vectores a través de la si-

militud coseno, se usó una implementación de *APSyn* basada en vectores, como se establece en Santus *et al.* [2016]. Está técnica está definida como la extensión de la intersección pesada entre los contextos más destacados de las palabras objetivo, ponderando por el rango promedio de las características cruzadas en las listas de contextos ordenados por *Positive Pointwise Mutual Information (PPMI)* de las palabras objetivo. Por lo tanto, el método usa algoritmos completamente diferentes para calcular la similitud entre dos vectores y debe proveer una perspectiva diferente en el comportamiento del modelo.

Datos	***n***	***Traslape***	**Dimensiones de vectores *wan2vec***						
			25	50	100	128	200	300	1000
MC-30	30	30	0.445	**0.805**	0.636	0.671	0.728	0.590	0.626
WS-353-SIM	204	183	0.538	0.530	0.532	0.608	**0.609**	0.594	0.532
MTurk-287	287	203	0.533	0.531	0.511	**0.565**	0.480	0.519	0.466
WS-353-REL	253	214	0.397	0.446	0.439	0.479	**0.496**	0.427	0.435
MEN-TR-3k	3000	272	0.550	**0.637**	0.615	0.605	0.614	0.578	0.551
SimLex-999	999	968	0.304	0.307	0.350	0.345	0.361	0.342	**0.396**
MTurk-771	771	698	0.492	0.528	0.535	0.514	**0.567**	0.534	0.533
RG-65	65	65	0.533	**0.760**	0.734	0.660	0.649	0.564	0.620
Promedio			0.443	0.515	0.515	0.521	**0.533**	0.495	0.494

Tabla 6-24: Correlación de Spearman entre las predicciones de *Wan2vec* (basados en *APSyn*) y los conjuntos de referencia. El grafo fue construido usando ϕ_{FI} como función de peso.

La tablas 6-24 y 6-25 muestran los resultados de *APSyn* con los mismos parámetros de comparación que se probaron anteriormente y con las mismas funciones de peso ϕ_{FI} y ϕ_{FAI}. Se observa fácilmente que, cuando se comparan con 6-22 y 6-23, los resultados son mucho más bajos que este método, sin embargo, en general, ϕ_{FAI} alcanza mejores resultados que ϕ_{FI}. Tomando los promedios, se puede observar que las dimensiones 50 y 200 funcionan mejor con ϕ_{FI}, mientras que los otros alcanzan mejores puntuaciones con ϕ_{FAI}. Eso no es exactamente lo que sucede con la similitud coseno, que devolvió mejores resultados con 1000 dimensiones con ϕ_{FI} y con 200 dimensiones con ϕ_{FAI}.

Respecto a los conjuntos de referencia, los mejores resultados son aún los obtenidos con la

Datos	***n***	***Traslape***	**Dimensiones de vectores *wan2vec***						
			25	50	100	128	200	300	1000
MC-30	30	30	0.460	0.682	0.607	0.687	0.681	0.674	**0.741**
WS-353-SIM	204	183	0.580	0.586	**0.638**	0.630	0.558	0.624	0.574
MTurk-287	287	203	**0.638**	0.551	0.562	0.588	0.524	0.533	0.354
WS-353-REL	253	214	0.457	0.443	**0.516**	0.487	0.444	0.499	0.478
MEN-TR-3k	3000	272	0.590	0.626	0.634	**0.635**	0.601	0.602	0.569
SimLex-999	999	968	0.301	0.335	0.369	0.363	0.368	0.372	**0.405**
MTurk-771	771	698	0.524	0.492	0.525	**0.559**	0.524	0.529	0.552
RG-65	65	65	0.512	0.596	0.671	**0.685**	0.574	0.590	0.633
Promedio			0.471	0.489	0.522	**0.535**	0.488	0.520	0.517

Tabla 6-25: Correlación de Spearman entre las predicciones de *Wan2vec* (basados en *APSyn*) y los conjuntos de referencia. El grafo fue construido usando ϕ_{FAI} como función de peso.

lista MC-30, pero esta vez con ϕ_{FI} y dimensión 50. A pesar que para ϕ_{FAI} el mejor puntaje se alcanza con esta lista, la dimensión necesitada esta vez es de 1000. En general, 50 y 200 son las mejores dimensiones con ϕ_{FI} y 128 con ϕ_{FAI}. Este último resultado es consistente con la configuración por defecto del recurso *node2vec*.

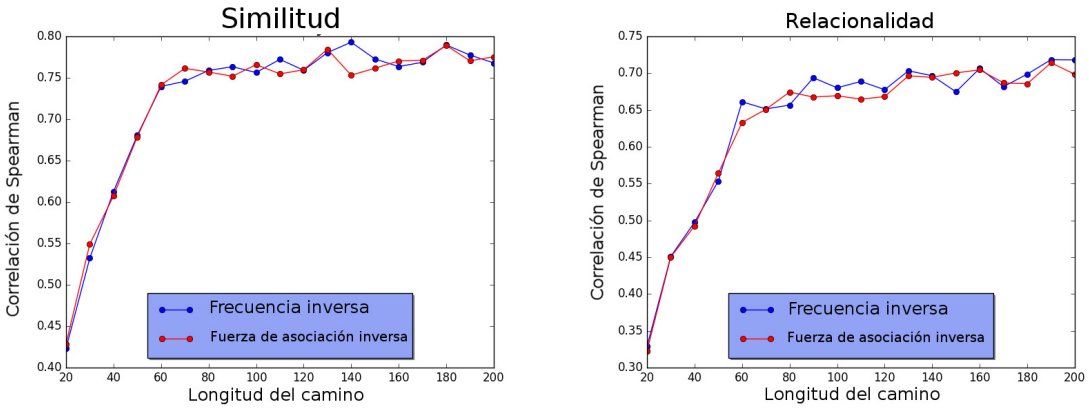

Figura 6-2: Correlaciones de Spearman obtenidos con diferentes longitudes de camino usando ambas funciones de pesado ϕ_{FI} and ϕ_{FAI}.

Análisis de la longitud del camino en *node2vec* Se realizó un análisis adicional que consiste en identificar la longitud de camino óptimo del algoritmo *node2vec*. La longitud del

camino indica qué tan profundo el algoritmo puede moverse en el grafo para obtener el vector correspondiente al nodo. El valor por defecto es 80 y ese es el que se usó en los experimentos de arriba.

Se midió el desempeño de los vectores de *Wan2vec* efectuando los mismos experimentos que la sección anterior, pero cambiando sistemáticamente la longitud del camino en el recorrido que el algoritmo *node2vec* realiza para mapear cada nodo. Se evaluó la longitud del camino desde 20 hasta 200 en intervalos de 10. La figura 6-2 presenta los resultados de los experimentos con dos conjuntos de datos, el WS-353-REL (Relacionalidad) y el WS-353-SIM (Similitud).

Modelo	**Corpus (tamaño)**	**Vocabulario**	**Autor**	**Arquitectura**	**Algoritmo**	**Tam. ventana de contexto**
word2vec	Google News (100B)	3M	Google	word2vec	muestreo negativo	BoW - 5
GloVe 6B	Wikipedia + Gigaword 5 (6B)	400,000	GloVe	GloVe	AdaGrad	10+10
GloVe 42B	Common Crawl (42B)	1.9M	GloVe	GloVe	GloVe	AdaGrad
fastText	Wikipedia	desconocido	Facebook	fastText	skip-gram	*n-gramas* de caracteres [1-5]

Tabla 6-26: Descripción de los vectores pre-entrenados de los tres modelos de *embeddings* evaluados. Todos ellos de dimensión 300.

En cada caso, el mejor resultado es alcanzado después de una longitud del camino de 60, alcanzando el mejor desempeño cerca de 120. Sin embargo, en este punto, el mejoramiento de la calidad de los vectores no es significativo y puede incrementarse el tiempo y complejidad de entrenar el modelo.

Comparación con modelos pre-entrenados Con la finalidad de probar y comparar la calidad de los vectores de *Wan2vec*, se realizaron experimentos con vectores pre-entrenados, se seleccionaron tres modelos vectoriales: *word2vec*[27], *GloVe*[28], y *fastText*[29].

La tabla 6-26 presenta las características generales de los recursos evaluados. En todos los casos, las dimensiones vectoriales es de 300. La diferencia de estos tres modelos radica

[27] `https://code.google.com/archive/p/word2vec/`
[28] `https://nlp.stanford.edu/projects/glove/`
[29] `https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md`

básicamente en el tipo y tamaño del corpus de entrenamiento, el número total de palabras diferentes, el algoritmo de entrenamiento, y el tamaño de la ventana de contexto. Los datos de los autores de estos modelos también son presentados.

Datos	**n**	**n(Traslape)**	**GloVe 6B**	**GloVe 42B**	**word2vec**	**fastText**	**Wan2vec-300**	**Wan2vec-100**
MC-30	30	30	0.702	0.777	0.788	0.811	**0.858**	0.791
WS-353-SIM	204	183	0.656	0.691	0.764	0.775	0.762	**0.777**
MTurk-287	287	203	0.707	0.718	0.726	**0.736**	0.666	0.697
WS-353-REL	253	214	0.604	0.621	0.632	0.688	0.658	**0.705**
MEN-TR-3k	3000	272	0.735	0.733	0.769	0.764	0.764	**0.807**
SimLex-999	999	968	0.370	0.373	0.441	0.376	**0.510**	0.490
MTurk-771	771	698	0.648	0.680	0.673	0.671	0.697	**0.730**
RG-65	65	65	0.769	0.817	0.760	**0.799**	0.791	0.798
Promedio			0.649	0.676	0.694	0.703	0.713	**0.724**

Tabla 6-27: Correlación de Spearman entre vectores pre-entrenados, *Wan2vec* con dimensión 300 y *Wan2vec* con su mejor valor de correlación, basados en similitud coseno.

La tabla 6-27 muestra la correlación de Spearman entre los modelos de vectores pre-entrenados y los parámetros de comparación. La columna *n(Traslape)* es la misma que en las tablas 6-22 a 6-25, de esta forma se realizó una evaluación justa usando únicamente las palabras que están disponibles en todos los modelos vectoriales (incluyendo *Wan2vec)*. Las últimas dos columnas muestran los resultados obtenidos con *Wan2vec*. Una devuelve las salidas del método con vectores de dimensión 300 con ϕ_{FAI} como función de peso. La última tiene el mejor puntaje, representado por los vectores con dimensión 100 y ϕ_{FAI} como función de peso.

En promedio, la similitud obtenida con los vectores *GloVe* obtuvo los peores puntajes de correlación, mientras que la similitud obtenida con *fastText* alcanzó 0.703 de la correlación de Spearman. Estas salidas fueron superadas por *Wan2vec*, ambas con dimensión 300 (0.713) y 100 (0.724). El rendimiento de *fastText* es mejor que *Wan2vec* con dimensión 300, con WS-353-SIM, MTurk-287, WS-353-REL, MEN-TR-3k y RG-65, mientras que los vectores *Wan2vec* con dimensión 300 superaron *fastText* con MC-30, SimLex-999 y MTurk-771. Sin embargo, tomando la mejor dimensión, 100, *Wan2vec* alcanza mejores resultados que *fastText*, excepto para MTurk-287 y RG-65. Con esos conjuntos de comparación, *fastText* obtiene siempre los mejores puntajes.

En cuanto a los vectores de *Wan2vec*, las correlaciones más altas fueron alcanzadas con el

Datos	**n**	**n(Traslape)**	**GloVe 6B**	**GloVe 42B**	**word2vec**	**fastText**	**Wan2vec 300**	**Wan2vec 128**
MC-30	30	30	0.635	0.727	0.739	**0.767**	0.674	0.687
WS-353-SIM	204	183	0.624	**0.688**	0.682	0.611	0.624	0.630
MTurk-287	287	203	0.591	**0.649**	0.541	0.627	0.533	0.588
WS-353-REL	253	214	0.444	0.593	**0.602**	0.453	0.499	0.487
MEN-TR-3k	3000	2720	0.550	0.605	0.629	0.587	0.602	**0.635**
SimLex-999	999	968	0.223	0.334	0.368	0.263	**0.372**	0.363
MTurk-771	771	698	0.508	0.581	**0.587**	0.488	0.529	0.559
RG-65	65	65	0.648	0.686	**0.748**	0.649	0.590	0.685
Promedio			0.528	0.608	**0.612**	0.556	0.520	0.535

Tabla 6-28: Correlación de Spearman entre vectores pre-entrenados, *Wan2vec* con dimensión 300 y *Wan2vec* con su mejor valor de correlación, basados en *APSyn*.

MC-30(300) y MEN-TR-3k (100), obteniendo correlaciones de 0.858 y 0.807, respectivamente. Los vectores pre-entrenados alcanzaron la más alta correlación de 0.811 en el MC-30 con los vectores *fastText* y 0.817 en RG-65 con *GloVe 42B*. También es interesante notar que la correlación con el corpus SimLex-999 es el más bajo de todos los modelos vectoriales, siendo el mejor el 0.510 de *Wan2vec*-300 y el 0.490 de *Wan2vec*-100, entre los otros métodos.

Las mismas pruebas se realizaron con la medida de similitud *APSyn*, con los resultados mostrados en la tabla 6-28. Las columnas siguen la misma estructura que la tabla 6-27. En general, los resultados son muy bajos, como sucedió con *Wan2vec* cuando se usó esta medida (tablas 6-25 y 6-24). Adicionalmente, la principal diferencia entre las tablas 6-27 y 6-28 es que, en la última, los vectores *GloVe* son mas competitivos. Ambos *GloVe 42B* y *word2vec* alcanzan mejores resultados que *fastText*. *Wan2vec* con dimensión 300 tiene resultados consistentes, obteniendo el mejor puntaje con SimLex-999. Finalmente, si se toma *Wan2vec* con la mejor dimensión, se obtienen los mejores puntajes con el conjunto MEN-TR-3k. Sin embargo, *GloVe 42B* gana con WS-353-SIM y MTurk-287, mientras que *word2vec* tiene los mejores resultados con WS-353-REL, MTurk-771 y RG-65. *FastText* alcanzó los mejores puntajes con MC-30.

Por lo tanto, es claro que la medida de similitud que se usó tiene un claro impacto en los resultados obtenidos. Los modelos que están siendo comparados muestran comportamientos similares bajo pruebas similares, mostrando un inesperado descenso del rendimiento bajo la medida *APSyn*. De hecho, se observa una inversión de los puntajes obtenidos usando la similitud coseno. *Wan2vec* y *fastText*, que trabajaron mejor con coseno, son los peores con *APSyn*,

mientras que *GloVe* y *word2vec*, los peores sistemas con coseno, obtuvieron mejores números en *APSyn*.

6.4.2.2. Evaluación extrínseca

Con la finalidad de evaluar *Wan2vec* en tareas de aplicación, se usó *SentEval*[30] [Conneau y Kiela, 2018], una librería para evaluar la calidad de vectores de oraciones (*sentence embeddings*). Con estas herramientas, se evaluará si los vectores de *Wan2vec* tienen capacidad de generalización, usándolos como entradas en varias tareas de transferencia. En particular, se incluyó en la evaluación:

- Tareas de análisis de sentimientos (SSTy-2, SST-5): ambas en binario y de ajuste fino *fine-grained Stanford Sentiment Treebank (SST)* [Socher *et al.*, 2013].
- Tarea de detección de paráfrasis (MRPC): tiene como objetivo identificar si un par de oraciones mantienen una relación de equivalencia semántica.
- Tarea de inferencia de lenguaje natural (SICK-E): consiste en predecir si dos oraciones de entrada están implicadas, neutrales o contradictorias [Marelli *et al.*, 2014].
- Tarea de similitud semántica textual (STS'12-16): consiste en evaluar cómo la distancia coseno entre dos oraciones se correlaciona con un puntaje de similitud etiquetado por humanos, a través de correlaciones de Pearson.

Para las tareas de clasificación, SentEval genera vectores de oraciones y un clasificador de regresión logística. Para las tareas no supervisadas (STS), también se generar vectores de oraciones desde los vectores de palabras. En ambos casos, los vectores de oración son el promedio de los vectores de cada palabra en cada oración. A pesar de que existen otros métodos universales de codificación de oraciones [Conneau *et al.*, 2017], el promedio de los vectores de palabras es una base competitiva para comparar la calidad de los vectores de palabras.

La tabla 6-29 muestra la evaluación de vectores con modelos pre-entrenados (*GloVe* y *fastText*) y los vectores *Wan2vec* en varias tareas de clasificación (SST, MRCP, y SICK-E). Los

[30] `https://github.com/facebookresearch/SentEval`

resultados *Wan2vec* están cerca de 10 puntos por debajo que los obtenidos por *GloVe* y *fastText*. Con la finalidad de determinar si esto se debe al vocabulario reducido de *Wan2vec*, se realizaron las pruebas con un conjunto léxico para *GloVe* y *fastText* equivalente al tenido en *Wan2vec*. Como se puede apreciar, los resultados ahora son comparables y para tareas MRPC y SICK-E, el sistema basado en *Wan2vec* con la función de peso ϕ_{FAI} obtuvo los mejores puntajes.

El experimento destaca que a pesar de la evaluación intrínseca el vocabulario reducido no era un problema, para las tareas contenidas en SentEval es una clara limitación.

Modelo	**SST-2**	**SST-5**	**MRPC**	**SICK-E**
GloVe 42B	79.01	45.23	72.62	79.01
fastText	**81.88**	**45.43**	**73.28**	**79.20**
GloVe (reducido)	73.31	37.19	70.55	76.72
fastText (reducido)	74.30	38.46	70.49	78.00
Wan2vec ϕ_{FAI}	70.40	37.29	70.90	77.21
Wan2vec ϕ_{FI}	70.35	38.19	71.36	78.08

Tabla 6-29: Resultados de tareas de transferencia para modelos vectoriales pre-entrenados y *Wan2vec*. Todos los vectores tienen dimensión 300.

La tabla 6-30 muestra la evaluación de modelos vectoriales pre-entrenados (*GloVe* y *fastText*) y los vectores de *Wan2vec* en los conjuntos de datos para medir similitud textual (STS'12, STS'13, STS'14, STS'15, STS'16). Las pruebas muestran el mismo comportamiento general que los anteriores. Con el vocabulario completo, *Wan2vec* no es competitivo comparado con *GloVe* y *fastText*, mientras que, con el mismo vocabulario, *Wan2vec* alcanza los mejores puntajes en cada tarea, excepto STS'13. Vale la pena señalar que siguiendo las tablas 6-29 y 6-30, la función de peso ϕ_{FI} parece trabajar mejor que ϕ_{FAI} para esta tarea.

SentEval también incluye una serie de tareas [31] [Conneau *et al.*, 2018] de prueba para evaluar las propiedades lingüísticas codificadas en una oración vectorial dada. La tarea está dividida en tres grupos, dependiendo del tipo de información codificada:

- Información superficial: evalúa la habilidad de las oraciones vectoriales para preservar

[31] `https://github.com/facebookresearch/SentEval/tree/master/data/probing`

Modelo	**STS'12**	**STS'13**	**STS'14**	**STS'15**	**STS'16**
GloVe 42B	0.5208	0.4960	0.5460	0.5607	0.5144
fastText	**0.5826**	**0.5790**	**0.6491**	**0.6760**	**0.6427**
GloVe (Reducido)	0.3528	0.2430	0.4413	0.4639	0.3719
fastText (Reducido)	0.3468	0.3504	0.4773	0.4879	0.4001
Wan2vec ϕ_{FAI}	0.3411	0.3503	0.4858	0.5227	0.4127
Wan2vec ϕ_{FI}	0.3471	0.3499	0.4840	0.5291	0.4135

Tabla 6-30: Evaluación de representaciones de oración en tareas de similitud textual. El promedio de las correlaciones de Pearson es usado para STS'12 a STS'16, los cuales están compuestos por varias subtareas. Todos lo vectores tienen dimensión 300.

las propiedades superficiales de la oración original. El objetivo de la tarea de longitud de la oración **SentLen** es predecir la longitud de la oración en términos del número de palabras. El objetivo de la tarea de contenido de la palabra **WC** es predecir cuál de las palabras objetivo aparece en las oración dada. De esta forma, evalúa si es posible recuperar información acerca de las palabras originales en la oración desde su representación vectorial.

- Información sintáctica: evalúa la habilidad de las oraciones vectoriales para preservar las propiedades sintácticas de las oraciones que codifican. El objetivo de la tarea denominada *cambio de bigrama* (**BShift**) evalúa si la oración vectorial puede distinguir el orden de las palabras en la oración. La tarea *profundidad del árbol* (**TreeDepth**) tiene como objetivo predecir la profundidad máxima de un árbol sintáctico de una oración. La tarea de *constituyentes superiores* (**TopConst**) prueba la capacidad de las oraciones vectoriales para capturar estructuras sintácticas latentes mediante la clasificación de la secuencia de constituyentes superiores.

- Información semántica: las tareas en este grupo requieren entender el significado de la oración. La tarea *tiempo* **(Tense)** tiene como objetivo detectar el tiempo verbal de la clausula principal (presente o pasado). La tarea de *número de sujeto* (**SubjNum**) tiene como objetivo predecir el número del sujeto de la cláusula principal (singular o plural). De

la misma forma, la tarea *número de objeto* (**ObjNum**) busca el número del objeto directo de la cláusula principal. La tarea del *extraño semántico* (**SOMO**) evalúa si una oración aparece tal cual en el corpus original, o si un (único) sustantivo o verbo elegido al azar fue reemplazado por otra forma con la misma categoría sintáctica. La tarea de inversión de coordinación (**CoordInv**) tiene como objetivo distinguir entre la oración original y las oraciones en las que el orden de dos conjunciones de cláusulas coordinadas se ha invertido.

La evaluación de las tareas probadas se presentan en la tabla 6-31. Otra vez, la tendencia general es que *Wan2vec* solo obtiene resultados comparables cuando *GloVe* y *fastText* usan la misma cantidad de vocabulario; y, en último caso, *fastText* siempre es ligeramente mejor que *Wan2vec*. Solo para la tarea **TopConst**, *Wan2vec* obtiene mejores resultados con ϕ_{FI}. Sin embargo, hay una excepción con **SOMO**, que parece no verse influenciado por la cantidad de vocabulario. En esa tarea, *Wan2vec* con ϕ_{FI} supera los otros sistemas. Asimismo, es también destacable la manera en que *GloVe*, cuando se reduce, tiene puntajes muy bajos con **SentLen**.

Modelo	**Sent-Len**	**WC**	**Tree-Depth**	**Top-Const**	**BShift**	**Tense**	**Subj-Num**	**Obj-Num**	**SOMO**	**Coord-Inv**
GloVe 42B	**53.79**	**90.86**	31.63	63.16	**50.04**	**84.03**	78.28	76.65	49.74	**54.21**
fasttext	50.48	92.05	**32.01**	**63.63**	49.74	86.59	**79.79**	**79.73**	49.74	53.17
GloVe (reducido)	27.16	70.72	25.25	32.75	49.61	70.52	72.09	73.50	49.11	51.85
fasttext (reducido)	47.06	71.95	29.36	35.87	49.92	70.56	72.98	74.60	49.94	51.74
wan2vec ϕ_{FAI}	43.78	69.51	29.18	36.38	49.26	68.66	70.12	71.74	49.77	51.46
wan2vec ϕ_{FI}	44.63	69.13	29.33	37.13	49.49	68.05	69.81	71.95	**50.31**	50.97

Tabla 6-31: Precisión de las tareas con regresión logística. Todos los vectores tienen 300 dimensiones.

Se hace notar que solamente se probó *Wan2vec* con 300 dimensiones, que no es la mejor configuración. Probablemente usar otras dimensiones que obtuvieran mejores resultados en la evaluación intrínseca, como la 100, el comportamiento de *Wan2vec* podría mejorar. A pesar de ello, como conclusión, la evaluación extrínseca ha demostrado que es necesario tener un vocabulario amplio para obtener resultados competitivos con todo tipo de tarea.

Consideraciones finales de *Wan2vec* Resumiendo el método presentado en esta sección, se mostró que a través de la aplicación de *node2vec* utilizando el corpus de Edinburgh Associative Thesaurus (EAT), se obtuvieron un conjunto de vectores que alcanzaron una mejor correlación con conjuntos de datos creados a través de anotadores humanos, que otros modelos vectoriales en la tarea de predicción de similitud y relacionalidad. Los vectores entrenados con el grafo usando la función de peso ϕ_{FAI} alcanzaron un mejor promedio en la correlación de Spearman, que los entrenados en el grafo con el peso ϕ_{FI}. Adicionalmente, se hizo un experimento utilizando una estrategia de reducción de nodos, que consiste en mantener únicamente aquellos nodos que están fuertemente conectados. Siguiendo el trabajo previo de De Deyne *et al.* [2016], se mantuvieron solo las respuestas que también ocurren como estímulos y los estímulos que también se dieron como respuestas. Lo que resultó en una gran pérdida de nodos y de un traslape reducido, lo cual impidió que se realizaran experimentos adicionales que sean fuertes como parámetros de comparación. Los resultados que se reportaron claramente superan los que se obtuvieron con algunos vectores pre-entrenados (*word2vec*, *GloVe*, FasText) que fueron entrenados con grandes volúmenes de información como *Wikipedia*. Los resultados están en línea con el trabajo de De Deyne *et al.* [2016], reafirmando la importancia de las Normas de Asociación de Palabras como un recurso para tareas de procesamiento de lenguaje natural. Este tipo de recursos reflejan, hasta cierto punto, la representación mental de las palabras.

Por el contrario, el peor desempeño del modelo ha sido reportado cuando se trata de evaluaciones extrínsecas. Las pruebas que se realizaron indican que el vocabulario reducido es la principal razón de estos resultados. Por lo tanto, este es uno de los principales problemas con los que tiene que lidiar este modelo.

6.4.3. Normas en Español

Para la evaluación de vectores en español utilizando el grafo construido con normas de asociación y aprendidos a través de *node2vec*, se tuvo una primera aproximación con las Normas de Asociación de Palabras para el español de México descritas en el capítulo 2, sección 2.5.1.1. Posteriormente, se complementaron las pruebas utilizando las Normas de Asociación Libres en

Castellano, descritas en la sección 2.5.1.2. Estas dos evaluaciones fueron reportadas en [Reyes-Magaña *et al.*, 2018] y [Gómez-Adorno *et al.*, 2019], respectivamente.

Para evaluar la similitud, primero se llevó a cabo una visualización de una muestra de palabras de NAP usando la proyección *t-SNE* de los vectores de palabras en un espacio vectorial bi-dimensional. En la figura 6-3 se aprecia cómo se agrupan las palabras que están relacionadas entre sí. Se muestran los resultados obtenidos con las tres formas de peso de lados, y se observa que todas son capaces de detectar algunas coincidencias en el significado. Las figuras ilustran algunos fenómenos interesantes. Por ejemplo, cuando se toma la frecuencia como peso, la palabra *pájaro* se dibuja muy cerca de *avión*. De aquí se infiere que la característica *volar* es más representativa que *animal* para el modelo. Por su parte, la palabra *caballo*, se representa más cercano a *camioneta* que a otros animales, incidiendo más en su condición de *medio de transporte*.

Además, se evaluó la capacidad de los vectores de palabras para capturar las relaciones semánticas mediante una tarea de similitud de palabras. Específicamente, se usó el subconjunto (150 pares de palabras) del corpus WordSim-353 [Finkelstein *et al.*, 2001] compuesto por pares de términos semánticamente relacionados con puntuaciones de similitud dadas por humanos. Hassan y Mihalcea [2009] elaboraron una versión de este corpus en español [32].

Se calculó la similitud coseno entre los vectores del subconjunto de pares de palabras contenidos en el corpus WordSim-353 y se comparó con la similitud dada por humanos. Las tablas 6-32 y 6-33 presentan la correlación de Spearman, en porcentajes, de la similitud dada por etiquetadores humanos, con la similitud obtenida con vectores de palabras (aprendidos del NAP) de diferentes dimensiones aprendidos en los grafos dirigidos y no dirigidos, respectivamente.

Se puede observar que los vectores que se obtienen con los grafos dirigidos no son capaces de trasladar los vecindarios de los nodos al espacio vectorial. En cambio, a causa de la naturaleza no restringida de los lados, el algoritmo *node2vec* es capaz de caminar diferentes vecindarios en el grafo no dirigido, y por ello consigue mejores representaciones vectoriales.

La tabla 6-34 muestra la correlación de Spearman entre la similitud coseno obtenida con

[32] `http://web.eecs.umich.edu/~mihalcea/downloads.html`

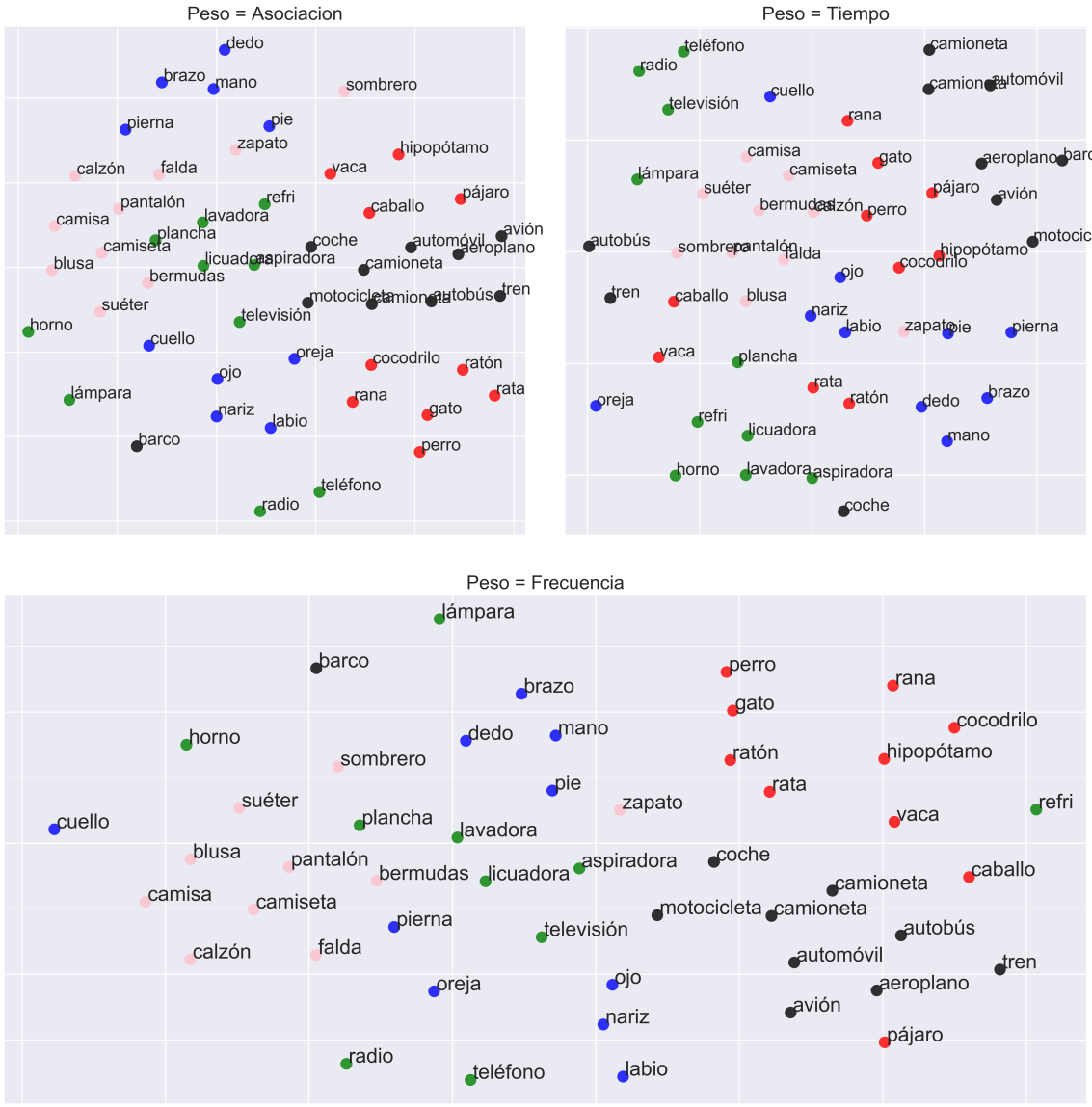

Figura 6-3: Proyección de los vectores de palabras para 5 grupos semánticos (de diez palabras cada uno). Los colores están codificados como sigue: animales - rojo, transporte - negro, partes del cuerpo - azul, electrodomésticos - verde y ropa - rosa.

los vectores pre-entrenados de *word2vec* y la similitud de los humanos (obtenida del corpus WordSim-353). El valor de correlación más alto fue obtenido con los vectores entrenados en el *Spanish Billion Word Corpus* [Cardellino, 2016] (w2v-1b). Los vectores entrenados con *Wikipedia*[33] en español (w2v-wk) obtuvieron resultados similares a los del método. Los mejores

[33]Vectores de palabras de más de 30 lenguajes: `https://github.com/Kyubyong/wordvectors`

Tabla 6-32: Correlación de Spearman (%) de la similitud coseno calculada con vectores obtenidos del grafo dirigido.

Tamaño del vector	**Frecuencia**	**Asociación**	**Tiempo**
300	-3.07	-3.11	-3.11
200	-1.95	-1.99	-2.03
128	0.88	0.98	0.96
100	**4.61**	**4.61**	**4.63**
50	2.51	2.42	2.39
25	-3.79	-3.89	-3.92

Tabla 6-33: Correlación de Spearman (%) de la similitud coseno calculada con vectores obtenidos del grafo no dirigido.

Tamaño del vector	**Frecuencia**	**Asociación**	**Tiempo**
300	43.62	43.58	50.77
200	42.89	40.55	44.67
128	39.54	44.01	50.31
100	44.66	44.31	46.50
50	45.60	**47.52**	**53.42**
25	**47.71**	45.75	51.04

resultados con los vectores entrenados con *node2vec* basados en el NAP se registraron con el grafo no dirigido, considerando el *tiempo* como medida de peso en los lados del grafo.

Tabla 6-34: Comparación con vectores pre-entrenados.

Fuente	**Tamaño del vector**	**Correlación de Spearman**
w2v-1b	300	**62.20**
w2v-wk	300	53.37
nap-tiempo	300	50.77
nap-tiempo	50	53.42

Visto desde la perspectiva del funcionamiento del sistema *node2vec*, esto no debería ser una sorpresa. Las palabras con un índice más alto de asociación normalmente tienen un tiempo de

formulación más breve, y el algoritmo busca los caminos más cortos. Este experimento permite visualizar ajustes a las variables de frecuencia y fuerza asociativa para obtener resultados más concluyentes.

Los resultados reportados son comparables a los obtenidos con *word2vec* entrenado con grandes corpus. El rendimiento incluso mejora los resultados alcanzados con *word2vec* entrenados en *Wikipedia*.

Las evaluaciones realizadas con los vectores generados con el corpus NAP mostraron resultados prometedores respecto a los índices de similitud y relacionalidad.

Con el objetivo de mejorar los resultados presentados anteriormente, se realizó el mismo experimento, pero esta vez usando las normas de asociación libres en castellano (NALC). Una de las principales diferencias comparadas con NAP radica en las funciones de peso, ya que NALC solamente proporciona la frecuencia de asociación con respecto al estímulo y con base en ella se calculó la fuerza de asociación. Asimismo, se ajustaron los pesos en los grafos usando como base ambas normas de asociación, permitiendo obtener los pesos menores que garanticen aquellos valores más representativos de manera adecuada para la ejecución de *node2vec*. Otra diferencia significativa respecto al NAP, es que las NALC tienen casi 6 mil estímulos, cuando en el NAP se tienen únicamente 234.

Para tratar la falta de vocabulario en cada conjunto de pares de palabras en las normas de asociación, se utilizó el concepto de traslape en los experimentos y se calculó el número de palabras comunes entre las listas que se comparan. Las otras son excluidas de la evaluación. Las tablas 6-39 y 6-40 presentan la correlación de Spearman, de la similitud dada por etiquetadores humanos, con la similitud obtenida con los vectores generados, comparando con NAP y NALC de manera separada. También se muestran diferentes dimensiones de vectores de palabras aprendidos a partir del grafo no dirigido con las diferentes funciones de peso. Se reportan el número de palabras que pueden ser encontradas en ambas, el corpus de normas de asociación (NAP o NALC) y el conjunto de evaluación (ES-WS-353 o MC-30).

Se puede observar que los vectores de palabras obtenidos con el corpus NALC alcanzaron mejores correlaciones, que aquellos obtenidos con el corpus NAP en ambos *datasets*, ES-WS-353

Tabla 6-35: Correlación de Spearman entre vectores de palabras basados en normas de asociación y ES-WS-353.

	NAP			**NALC**
	Traslape 140			**Traslape 322**
Dimensión	**FI**	**FAI**	**Tiempo**	**FAI**
300	0.489	0.463	0.461	0.650
200	0.454	0.456	0.491	0.641
128	0.503	0.463	0.450	0.659
100	0.471	0.478	0.495	**0.664**
50	**0.523**	**0.503**	0.503	0.626
25	0.484	0.478	**0.572**	0.611

Tabla 6-36: Correlación de Spearman entre vectores de palabras basados en normas de asociación y MC-30

	NAP			**NALC**
	Traslape 11			**Traslape 27**
Dimensión	**FI**	**FAI**	**Tiempo**	**FAI**
300	0.305	**0.563**	0.545	0.837
200	0.468	0.381	0.263	**0.844**
128	**0.545**	0.272	0.300	0.767
100	0.336	0.418	0.372	0.806
50	0.527	0.509	0.272	0.814
25	0.454	0.400	**0.563**	0.788

y MC-30. La diferencia en los resultados puede ser explicada por el tamaño del corpus en ambas normas de asociación, el corpus NALC tiene un mayor traslape en ambos *datasets* que el corpus NAP.

Los nuevos vectores también se compararon con vectores pre-entrenados en español [34]. Se seleccionaron 3 modelos vectoriales: *word2vec*, *GloVe*, y *fastText*.

La tabla 6-37 muestra la correlación de Spearman entre la similitud coseno obtenida con vectores pre-entrenados en grandes corpus y la similitud de WordSim-353 y MC-30, en compa-

[34] `https://github.com/uchile-nlp/spanish-word-embeddings`

ración con la correlación entre los vectores basados en NAP y las similitudes de los anotadores humanos. De la misma forma, la tabla 6-38 muestra la misma comparación con vectores pre-entrenados y los basados en el corpus NALC.

La correlación más alta fue la obtenida con los vectores entrenados con el modelo *fastText*. Los vectores entrenados con *Wikipedia* en español obtuvieron los mejores resultados entre los modelos pre-entrenados. Sin embargo, los resultados obtenidos con los vectores, tomando como base el corpus NALC, superaron los resultados con vectores pre-entrenados, en ambos *datasets* de evaluación, ES-WS-353 y MC-30.

Tabla 6-37: Correlación de Spearman de los vectores basados en NAP y vectores pre-entrenados respecto a los *datasets* de evaluación.

Fuente	**Tamaño del vector**	**MC-30 (Traslape 11)**	**ES-WS-353 (Traslape 140)**
Fasttext-sbwc	300	0.881	0.639
Fasttext-wiki	300	**0.936**	**0.701**
Glove-sbwc	300	0.827	0.532
Word2vec-sbwc	300	0.890	0.634
NAP-FAI	300	0.563	0.463
NAP-FI	300	0.305	0.489
NAP-Tiempo	25	0.563	0.572

Consideraciones finales sobre vectorización de NAP en español Esta sección presenta la obtención de vectores de palabras basados en NAP del español de México. Para ello, se inició a partir de un grafo construido con una pequeña colección con 4704 nodos, que corresponden al total de palabras disponibles en este recurso. La metodología aplicada es la misma que se hizo con *Wan2vec* para la obtención de vectores con el algoritmo *node2vec*.

Se observó que los experimentos muestran mejores resultados con grafos no-dirigidos. La realización de estas pruebas fueron las primeros que se realizaron cuando se trabajó con *node2vec* y los corpus de normas de asociación de palabras, esto permitió visualizar los ajustes

Tabla 6-38: Correlación de Spearman de los vectores basados en NALC y vectores pre-entrenados respecto a los *datasets* de evaluación.

Fuente	**Tamaño del vector**	**MC-30 (Traslape 27)**	**ES-WS-353 (Traslape 322)**
Fasttext-sbwc	300	0.762	0.613
Fasttext-wiki	300	0.793	0.624
Glove-sbwc	300	0.707	0.482
Word2vec-sbwc	300	0.795	0.624
NALC-FAI	300	0.837	0.650
NALC-FAI	200	**0.844**	**0.664**

necesarios sobre los pesos del grafo favoreciendo los caminos más importantes con el menor valor. Se observó que al utilizar las Normas de Asociación Libre en Castellano, se mejoraron los resultados, esto en gran medida por el incremento de vocabulario respecto a las NAP del español de México.

En cuanto a la validación extrínseca, se observó que con *Wan2Vec* se utilizó *SentEval* como herramienta principal para la realización de las pruebas, desafortunadamente estas herramientas tan completas solo están disponibles para el idioma inglés, lo cual impide la realización de este tipo de experimentos aplicados sobre las NAP entrenadas en español. No obstante, la búsqueda léxica en sí, podría considerarse una tarea de aplicación en Procesamiento de Lenguaje Natural que usa directamente el recurso NAP.

6.5. Evaluación de Normas Automáticas de Asociación de Palabras

Para medir la calidad de las NAAP descritas en la sección 2.5.3.1, se realizaron dos tipos de experimentos. De manera similar a las evaluaciones de vectores, la primera prueba permite determinar la representatividad de *embeddings* generados con las NAAP, los cuales fueron creados con *node2vec*, tratando de describir las similitudes de los anotadores humanos. El segundo

experimento es sobre el modelo de búsqueda del diccionario inverso, que usando las NAP del Español de México como corpus del grafo, mostró una precisión alta. Debido a esto, en este experimento se usarán las Normas Automáticas para construir el grafo y, aplicando la centralidad intermedia dentro del modelo, se mostrará su rendimiento. Es importante destacar que la centralidad intermedia fue la mejor medida para la tarea del diccionario inverso.

Cada uno de los experimentos realizados en esta sección serán comparados con los resultados de las NAP del español de México. Se seleccionaron estos resultados como parámetro de referencia porque el corpus NAP y el NAAP son recursos que usan la misma variante del español (mexicano).

6.5.1. Vectorización de las NAAP

Se utilizó *node2vec* para entrenar vectores de 300 dimensiones, el corpus para la construcción del grafo fueron las Normas Automáticas de Asociación de Palabras (NAAP). Con los vectores entrenados, se evaluó su habilidad para capturar las relaciones semánticas a través de la tarea de similitud descrita anteriormente. Se usaron los corpus de WordSim-353 y MC-30, ambos en español. Se calculó la similitud coseno entre los pares de palabras contenidos en los corpus y posteriormente se compararon con las similitudes dadas por humanos usando la correlación de Spearman. Para hacer frente a la no inclusión de cada palabra de los conjuntos de datos de prueba de las NAAP, también se usó también el traslape .

Las tablas 6-39 y 6-40 presentan las correlaciones de Spearman de las similitudes dadas por etiquetadores humanos comparadas con las similitudes obtenidas de los vectores aprendidos tanto de NAP clásico como NAAP. Se reporta el traslape, que es el número de palabras que puede ser encontrado en ambos, ya sean las normas de asociación (NAP o NAAP) y el corpus de evaluación (ES-WS-353 o MC-30).

Se puede observar que los vectores de palabras obtenidos con el corpus NAAP alcanzaron mejores correlaciones con las similitudes de humanos que los vectores obtenidos con el NAP clásico en ambos corpus, ES-WS-353 y MC-30.

Tabla 6-39: Correlación de Spearman entre normas de asociación de palabras (español de México) con vectores de 300 dimensiones y el corpus ES-WS-353.

Norma de asociación	Función de peso	Traslape	Correlación
NAP	FI	140	0.489
	FAI		0.463
	Tiempo		0.461
NAAP fastText	Aproximación Inversa	291	**0.595**
NAAP GloVe			0.555
NAAP Word2Vec			0.550
NAAP FastWiki			0.572

Tabla 6-40: Correlación de Spearman entre normas de asociación de palabras (español de México) con vectores de 300 dimensiones y el corpus MC-30.

Norma de asociación	Función de peso	Traslape	Correlación
NAP	FI	11	0.305
	FAI		0.563
	Tiempo		0.545
NAAP fastText	Approximación Inversa	22	0.747
NAAP GloVe			0.698
NAAP Word2Vec			0.706
NAAP FastWiki			**0.771**

6.5.2. Modelo de búsqueda léxica y NAAP

El modelo de búsqueda léxica basado en centralidad intermedia utiliza la versión de subconjuntos de nodos [Brandes, 2008]. Se construyó el grafo NAAP considerando únicamente los 234 estímulos del NAP clásico, pero teniendo las respuestas y sus pesos propios del NAAP. En el Algoritmo 1 del capítulo 2, se describió el proceso para obtener dichas respuestas. El corpus de prueba utilizado es el mismo que se usó en la experimentación del modelo de búsqueda léxica para las NAP clásicas, disponible en el Anexo A.

Los resultados se muestran en la tabla 6-41. Es claro que cuando el modelo busca sobre los grafos con las NAP clásicas con cualquier función de peso, los resultados son más altos

que cuando se busca con el grafo NAAP. Se afirma que la precisión obtenida con el corpus NAAP es competitiva, debido a que los métodos de búsqueda de información con los que se comparó el modelo de búsqueda léxica aplicados al diccionario inverso, fueron todos superados por el grafo NAAP. Los métodos fueron: búsqueda de información boolena, *OneLook reverse dicionary*, *BM-25* [Robertson y Zaragoza, 2009], *CAS* [Ghosh *et al.*, 2014] y WantWords [Qi *et al.*, 2020].

Tabla 6-41: Resultados de búsqueda léxica en términos de precisión.

Normas de asociación	**Función de peso**	**p@1**	**p@3**	**p@5**
NAP	FI	0.616	0.741	0.774
	FAI	**0.655**	**0.804**	**0.829**
	Tiempo	0.362	0.550	0.652
NAAP fastText	Aproximación inversa	0.329	0.526	0.584
NAAP GloVe		0.333	0.544	0.587
NAAP Word2Vec		0.340	0.537	0.584
NAAP FastWiki		0.326	0.526	0.580

Se realizaron experimentos adicionales para podar el grafo. Para este fin, en cada NAAP se varió el peso con intervalos de incrementos de 0.05.

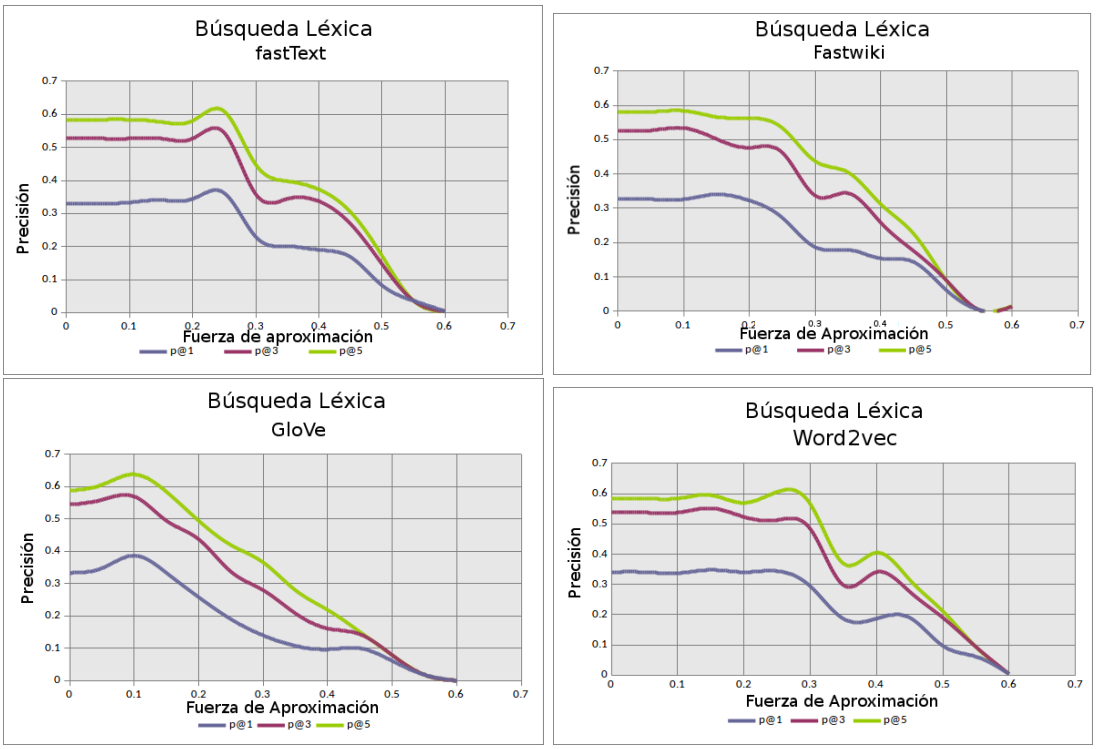

Figura 6-4: Precisión de la búsqueda léxica basada en NAAP

La figura 6-4 presenta la precisión de la búsqueda léxica; este valor se puede apreciar en el eje vertical. El eje horizontal representa de izquierda a derecha las reducciones de las respuestas que satisfacen el filtro, es decir, si se tiene un valor de 0.1, las respuestas a ser consideradas serán aquellas cuyos pesos varían de 1 a 0.1. En el caso de 0.55, se seleccionarán las respuestas con peso de 1 a 0.55, y así sucesivamente. Con esta técnica, se puede observar si hay alguna mejora en la precisión conforme se varían los valores en los pesos. La razón para realizar este experimento es que en algunos casos un grafo más compacto produce búsquedas más eficientes. Cuando la reducción alcanza el valor de 0.60 se tiene una mayor cantidad de palabras filtradas, lo que provoca menos palabras con las cuales trabajar, haciendo que la búsqueda léxica alcance valores cercanos a 0. Se puede observar que en los primeros intervalos, reducir el grafo no hace una diferencia significativa en la precisión. Se puede alcanzar un ligero pico antes de que la precisión comience a disminuir. Por esta razón, las NAAP se proporcionan sin ninguna reducción.

Consideraciones finales sobre NAAP A pesar que se pudo usar cualquier diccionario general del español como el de la Real Academia de la Lengua (RAE), el DEM presenta definiciones que se adecúan mejor al español de México, lo cual repercute en la evaluación del modelo y el tipo de corpus utilizado para este último fin. Sin embargo, la metodología propuesta para la construcción de las NAAP puede ser aplicada a cualquier tipo de diccionario.

Se siguió la misma forma de evaluación que se hizo con los *embeddings* basados en normas de asociación. Sin embargo, el objeto de prueba en este caso son las normas automáticas. En la evaluación intrínseca, se observó que los vectores generados a partir de las NAAP superan la correlación de *Spearman* de aquellos vectores que fueron entrenados con la NAP clásica.

En lo referente a la evaluación extrínseca, las pruebas presentan un uso más realista del corpus. La búsqueda léxica muestra que a pesar de no haber superado los resultados de la NAP clásica, la norma es lo suficientemente competitiva por haber superado modelos de recuperación de información clásicos. La metodología propuesta es una herramienta útil para crear normas de asociación de palabras y los elementos para producirlas son relativamente sencillos de obtener, que consisten principalmente de un diccionario y de vectores de palabras pre-entrenados. Las normas recolectadas con métodos clásicos producirán mejores resultados dependiendo de la tarea donde serán usados. Sin embargo, en algunos casos, donde el tiempo y disponibilidad de una norma de asociación de palabras es urgente o simplemente imposible de recolectar por medio de métodos clásicos, la creación de NAAP es una solución confiable y rápida. En una etapa más avanzada, el éxito de la técnica puede hacer innecesario el esfuerzo y los recursos que actualmente se dedican para recolectar normas de asociación de palabras.

Adicionalmente, como resultado paralelo, se proporcionan los vectores de palabras[35] que se entrenaron usando el algoritmo *node2vec*, teniendo como principal característica que estos vectores están basados en el español de México. Se afirma que con esta metodología se pueden producir vectores de palabras para variantes de un idioma sin la necesidad de una cantidad gigante de datos. Como trabajo a futuro, se pueden realizar experimentos adicionales para incrementar la precisión de la búsqueda léxica, como la aplicación de filtros en las palabras

[35] `https://drive.google.com/drive/folders/1nmApEvi4ywQl1CDjK5umiSQE79MuGP9C`

respuestas, conservando únicamente sustantivos, verbos y/o adjetivos. Roth y Im Walde [2008] mostraron que las NAP pueden ser enriquecidas usando diferentes tipos de corpus. Por lo que, más adelante, se pueden adicionar corpus enciclopédicos y de co-ocurrencias para mejorar el rendimiento de las tareas con NAP.

Capítulo 7

Conclusiones y trabajo futuro

En este capítulo se presentan las consideraciones finales del trabajo de investigación. Se dividen en 2 partes: a) aportaciones generales del trabajo de investigación y b) trabajo futuro.

7.1. Aportaciones

Este trabajo de investigación presenta un modelo de búsqueda léxica para la creación de un diccionario inverso, que desde la perspectiva de la lexicografía podría considerarse un diccionario onomasiológico, ya que el objetivo que se persigue es el mismo: encontrar una palabra objetivo a partir de diversos elementos que ayuden en el proceso de búsqueda. Se pudo observar que han existido muchas formas de resolver este problema, comenzando con soluciones físicas que permiten revisar libros que por medio de su organización van conduciendo a la palabra deseada. Claramente esto tiene varias desventajas, como la falta de las palabras adecuadas que conduzcan por el camino correcto hacia el objetivo, o en el caso de los diccionarios pictóricos, el hecho que la mayoría de los conceptos plasmados tienen que ser elementos no abstractos. Sin embargo, con el paso del tiempo y tomando ventaja del desarrollo tecnológico del mundo moderno se han podido desarrollar herramientas innovadoras dentro del ámbito de la lingüística computacional, que permitan dar solución a esta tarea.

De todas las propuestas tecnológicas que se revisaron durante el desarrollo de la tesis, nin-

guna tenía una aproximación como la que se implementó en este modelo de búsqueda léxica. El modelo que propone esta investigación tiene como una de las principales ventajas su simplicidad, el uso de técnicas basadas en grafos y el corpus sobre el cual está construido, estableciendo una forma novedosa del uso de este tipo de recursos psicolingüísticos. Las NAP permiten conocer la relación existente entre las palabras; aunque no están clasificadas explícitamente como sinonimia, hiperonimia, hiponimia, metonimia, etc., su relación está presente en las respuestas asignadas a un estímulo. Esta asociación estímulo-respuesta permite vislumbrar a las NAP como un conjunto de nodos con sus lados en una estructura tipo grafo. El reto consistía en poder encontrar aquellas palabras objetivo por medio de otras palabras a su alrededor. Es aquí donde hacen su aparición las medidas de centralidad de nodos, aquellas que nos permiten identificar los nodos más importantes del grafo, es así que se hizo un análisis exhaustivo de este tipo de medidas para el modelo de búsqueda léxica; realizando las adecuaciones del grafo original cuando así fue necesario. De todas las medidas utilizadas, dos obtuvieron los mejores resultados: CI y CC. Los demás modelos tuvieron resultados por debajo del BM25, que fue el más preciso de todos los clásicos que se probaron. Es importante notar que una ventaja adicional que se tiene con el corpus NAP son los valores numéricos entre las palabras; de manera particular, el valor de Fuerza de Asociación resultó ser el más adecuado para trabajar con los algoritmos explorados, que si bien no todos los NAP lo traen, puede ser calculado a partir de la frecuencia. Estos valores son traducidos como los pesos de los lados del grafo, haciendo más eficiente el proceso de búsqueda porque recalca la importancia de la relación entre dos nodos específicos. Con base en lo descrito anteriormente y en las pruebas realizadas, se afirma que sí es posible construir un modelo preciso de búsqueda léxica utilizando técnicas basadas en grafos. Asimismo, se presentó cómo las descripciones de conceptos, que son realizadas por gente común y corriente, con especificaciones no científicas, puede devolver resultados adecuados usando el modelo. Esto es posible gracias a la naturaleza del corpus. Las normas de asociación de palabras agrupan las palabras que se encuentran relacionadas de manera cognitiva, lo que nos lleva a pensar que no es necesario tener un gran dominio de la palabra objetivo, basta con tener una idea general que aborde algunos elementos relacionados con lo que se está buscando. Es evidente que mientras

mayor información se proporcione sobre el concepto, permitirá al algoritmo mejorar sus resultados. El éxito del sistema con definiciones no específicas, puede introducir nuevas líneas de investigación aplicada, y la implementación de diferentes asistentes de escritura especialmente orientados a personas con algún rango de afasias, como disnomia o Alzheimer.

Cuando se trabajó con las NAP del español de México, se observó que su principal problema es la cantidad restringida del número de palabras que pueden participar en la búsqueda. Para poder solucionar este problema, la tesis también presenta una metodología que permite la creación de normas de asociación automáticas de palabras.

De manera adicional, el corpus NAP fue abordado desde otra perspectiva, por medio de la generación de vectores de palabras o *word embeddings*. Las pruebas realizadas con los vectores generados a partir de corpus de normas de asociación mostraron resultados exitosos, unos en mayor medida que otros. Aunque los *embeddings* no forman parte del modelo de búsqueda léxica de manera directa, sí formaron parte de la investigación durante el desarrollo de la tesis. Y se consideran un aporte significativo al área de PLN porque el uso de vectores de palabras, hoy día impacta positivamente en el desarrollo de los nuevos alcances en el mundo de la lingüística computacional. Permiten obtener representaciones vectoriales de palabras que son comparables con aquellos entrenados en corpus de billones de palabras, pero sin la necesidad de tantos datos. Esto abre un área de oportunidad en la obtención de este tipo de representaciones para lenguas con bajos recursos digitales.

Finalmente, durante el desarrollo de la tesis se realizaron varias publicaciones que fueron mostrando el avance de la investigación.

Revistas

- *Research in Computing Science*. Artículo: Representaciones vectoriales de palabras de un corpus de normas de asociación de palabras [Reyes-Magaña *et al.*, 2018].

- *Journal of Intelligent Fuzzy Systems (JIFS)* (Factor de impacto 1.426). Artículo: A Lexical Search Model Based on Word Association Norms [Reyes-Magaña *et al.*, 2019a].

- *Semantic Web – Interoperability, Usability, Applicability an IOS Press Journal* (Factor de impacto 2.224). Artículo: Wan2Vec: Embeddings Learned on Word Association Norms [Bel-Enguix *et al.*, 2019].

Congresos

- Alberto Mendelzon International Workshop on Foundations of Data and Management. Artículo: Spanish Word Embeddings Learned on Word Association Norms [Gómez-Adorno *et al.*, 2019].

- ELEX 2019: SMART LEXICOGRAPHY. Artículo: Designing an Electronic Reverse Dictionary Based on Two Word Association Norms of English Language [Reyes-Magaña *et al.*, 2019b].

- Cognitive Aspects of the Lexicon. Artículo: Automatic Word Association Norms (AWAN) [Reyes-Magaña *et al.*, 2020].

Cabe señalar que se realizó una versión en línea del diccionario inverso, que está disponible tanto para español como inglés[1]. El algoritmo con el que funcionan ambos idiomas es el modelo basado en centralidad intermedia. El grafo de inglés está construido con las normas de *South Florida Free Association Norms*. Para la versión en español está construido actualmente con las *Normas de Asociación de Palabras para el español de México*.

7.2. Trabajo futuro

Se pudo observar que cuando se tienen normas de asociación de mayor tamaño, el rendimiento de la búsqueda léxica decrece. Debido a esto, es necesario encontrar métodos adicionales para acotar la búsqueda. Se considera que esta disminución es debido a la cantidad de caminos nuevos que se generan en el grafo, lo cual provoca que existan mayores posibilidades de transitar en los caminos más cortos. Se realizaron diversas formas para lograr este objetivo,

[1] `www.describe.com.mx/describeme`

una de las que mostraron mejor comportamiento fue solamente considerar un número limitado de respuestas asociadas a los estímulos, es decir, seleccionando aquellas que estuvieran mejor posicionadas después de la construcción completa del grafo. Así, se limitaba solamente a los nodos más significativos en su relación entre las palabras. Sin embargo, los resultados aún no son concluyentes. Se pueden explorar otras alternativas como limitar las palabras que forman parte de las respuestas realizando filtrados de solo sustantivos o adjetivos, incluso verbos; con su correspondiente análisis que pueda determinar la eficacia del método.

En lo referente a las normas de asociación automáticas, se puede medir su eficacia a través de otro tipo de pruebas; algunos estudios hacen análisis del tipo de relación semántica que capturan las normas automáticas, de esta forma es posible medir el porcentaje de relaciones semánticas con respecto a las recolectadas de manera clásica. En este caso se tendría que hacer un análisis mucho más exhaustivo para realizar la clasificación adecuada de cada una de las palabras respuesta obtenidas. Asimismo, se puede complementar el corpus base que sirve para la construcción de las normas automáticas, a través de recursos enciclopédicos o basados en corpus de co-ocurrencias. La mejora del modelo de búsqueda léxica también impactará en el desempeño de la tarea extrínseca utilizada para validar las NAAP. Finalmente, esto repercutirá también en el desempeño de la versión en línea del diccionario inverso para el español.

Bibliografía

Diccionario del Español de México (DEM). El Colegio de México, A.C. (2010)

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., y Soroa, A. A study on similarity and relatedness using distributional and WordNet-based approaches. En *Proceedings of NAACL-HLT 2009*, págs. 19–27. Association for Computational Linguistics, Stroudsburg, PA, USA (2009)

Aitchison, J. *Words in the mind: an introduction to the mental lexicon.* 4ª edición. Wiley-Blackwell (2012)

Al-Taie, M.Z. y Kadry, S. *Python for graph and network analysis.* Springer (2017)

Algarabel, S., Ruíz, J.C., y Sanmartín, J. *The University of Valencia's computerized Word pool.* Behavior Research Methods, Instruments & Computers. (1998)

Alvar Esquerra, M. Diccionarios ideológicos. *Libros* **24**:14–18 (1984)

Alvar Ezquerra, M., Clark, M., Mohan, B. *et al. Oxford-Duden pictorial Spanish and English dictionary.* Clarendon Press (1995)

Argota Vega, L.E., Reyes-Magaña, J., Gómez-Adorno, H., y Bel-Enguix, G. MineriaUNAM at SemEval-2019 Task 5: Detecting Hate Speech in Twitter using Multiple Features in a Combinatorial Framework. En *Proceedings of the 13th International Workshop on Semantic Evaluation*, págs. 447–452. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019)

Arias-Trejo, N., Barrón-Martínez, J.B., Alderete, R.H.L., y Aguirre, F.A.R. *Corpus de normas de asociación de palabras para el español de México [NAP].* UNAM, Universidad Nacional Autónoma de México. (2015)

Bahdanau, D., Cho, K., y Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014)

Bajo, M.T. Semantic facilitation with pictures and words. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **14**(4):579 (1988)

Baldinger, K. *Teoría semántica: hacia una semántica moderna*, tomo 12. Alcalá. (1970)

Baldinger, K. y Wright, R. *Semantic theory: towards a modern semantics.* B. Blackwell (1980)

Barcia, R. *Diccionario de sinónimos.* Colofón (2000)

Baroni, M., Dinu, G., y Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, tomo 1, págs. 238–247 (2014)

Bavelas, A. A mathematical model for group structure. *Social networks: critical concepts in sociology, New York: Routledge* **1**:161–88 (2002)

Bel-Enguix, G., Rapp, R., y Zock, M. A Graph-Based Approach for Computing Free Word Associations. En *Proceedings of the 9th edition of the Language Resources and Evaluation Conference*, págs. 221–230. (2014)

Bel-Enguix, G., Gómez-Adorno, H., Reyes-Magaña, J., y Sierra, G. Wan2vec: Embeddings learned on word association norms. *Semantic Web* **10**(6):991–1006 (2019)

Benot, E. *et al.* Diccionario de ideas afines y elementos de tecnología (1895)

Bernstein, T.M., Wagner, J. *et al. Bernstein's reverse dictionary.* Times Books (1975)

Bilac, S., Watanabe, W., Hashimoto, T., Tokunaga, T., y Tanaka, H. Dictionary search based on the target word description. En *Proceedings of the Tenth Anual Meeting of the Assocviation for Natural Language Processing*, págs. 556–559. (2004)

Bird, S. y Loper, E. NLTK: the natural language toolkit. En *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, pág. 31. Association for Computational Linguistics (2004)

Bloomfield, L. A set of postulates for the science of language. *Language* **2**(3):153–164 (1926)

Bojanowski, P., Grave, E., Joulin, A., y Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**:135–146 (2017)

Bonacich, P. Power and centrality: A family of measures. *American journal of sociology* **92**(5):1170–1182 (1987)

Bonin, P. *Mental lexicon: some words to talk about words*. Nova Science Publishers. (2004)

Borge-Holthoefer, J. y Arenas, A. Navigating Word Association Norms to Extract Semantic Information. En *Proceedings of the 31st Annual Conference of the Cognitive Science Society.*, tomo 621, págs. 2777–2782 (2009)

Brandes, U. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks* **30**(2):136–145 (2008)

Brandes, U. y Fleischer, D. Centrality measures based on current flow. En *Annual symposium on theoretical aspects of computer science*, págs. 533–544. Springer (2005)

Brath, R. y Jonker, D. *Graph analysis and visualization: discovering business opportunity in linked data*. John Wiley & Sons (2015)

Brown, A.S. A review of the tip-of-the-tongue experience. *Psychological bulletin* **109**(2):204 (1991)

Brown, A.S. *The tip of the tongue state.* Taylor & Francis (2012)

Bruni, E., Boleda, G., Baroni, M., y Tran, N.K. Distributional semantics in technicolor. En *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, págs. 136–145. Association for Computational Linguistics (2012)

Calero Medina, C.M. *Links in science: linking network and bibliometric analyses in the study of research performance.* Tesis Doctoral, Centre for Science Technology Studies (CWTS) , Faculty of Social and Behavioural Sciences , Leiden University (2012)

Cañs, J.J. Associative strength effects in the lexical decision task. *The Quarterly Journal of Experimental Psychology* **42**(1):121–145 (1990)

Cardellino, C. Spanish Billion Words Corpus and Embeddings (2016)

Casares, J. *Diccionario ideológico de la lengua española.* Gustavo Gili (1942)

Cleverdon, C. Optimizing Convenient Online Access to Bibliographic Databases. *Information services and Use* **4**:37–47 (1984)

Conneau, A. y Kiela, D. SentEval: An evaluation toolkit for universal sentence representations. En *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association (2018)

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., y Bordes, A. Supervised learning of universal sentence representations from natural language inference data. En *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, págs. 670–680. Association for Computational Linguistics (2017)

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., y Baroni, M. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. En *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, págs. 2126–2136. Association for Computational Linguistics (2018)

Davis, P.M. Cognición y aprendizaje: reseña de investigaciones realizadas entre grupos etnolingüísticos minoritarios (2014)

De Deyne, S., Perfors, A., y Navarro, D.J. Predicting human similarity judgments with distributional models: The value of word associations. En *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, págs. 1861–1870 (2016)

Devlin, J., Chang, M.W., Lee, K., y Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)

Dijkstra, E.W. *et al.* A note on two problems in connexion with graphs. *Numerische mathematik* **1**(1):269–271 (1959)

Doddington, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. En *Proceedings of the second international conference on Human Language Technology Research*, págs. 138–145. Morgan Kaufmann Publishers Inc. (2002)

Dutoit, M. y Nugues, P. A Lexical Database and an Algorithm to Find Words from Definitions. En *Proceedings of the 15th European Conference on Artificial Intelligence*, págs. 450–454. (2002)

Edmonds, D. *The Oxford reverse dictionary.* Oxford University Press, USA (2002)

El-Kahlout, I. y Oflazer, K. Use of Wordnet for Retrieving Words from Their Meanings. En *2nd Global WordNet Conference.* (2004)

Erkan, G. y Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* **22**:457–479 (2004)

Estrada, E. y Hatano, N. Communicability in complex networks. *Physical Review E* **77**(3):036111 (2008)

Estrada, E., Higham, D.J., y Hatano, N. Communicability betweenness in complex networks. *Physica A: Statistical Mechanics and its Applications* **388**(5):764–774 (2009)

Fajardo Uribe, L.A. La lingüística cognitiva: principios fundamentales. *Cuadernos de Lingüística Hispánica* (9):63–82 (2007)

Fernández, A., Diez, E., y Alonso, M. Normas de Asociación libre en castellano de la Universidad de Salamanca [Base de datos online]. *Recuperado de www.usal.es/gimc/nalc* (2010)

Fernández, A., Diez, E., y Alonso, M. Normas de falso recuerdo y reconocimiento (2015)

Fernández, A., Diez, E., Alonso, M.A., y Beato, M.S. Free-association norms form the Spanish names of the Snodgrass and Vanderwart pictures. *Behavior Research Methods, Instruments & Computers* **36**:577–583. (2004)

Ferrer i Cancho, R. y Solé, R. The small-world of human language. *Proceedings of the Royal Society of London. Series B, Biological Sciences* **268**(1482):2261–2265 (2001)

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., y Ruppin, E. Placing search in context: The concept revisited. En *Proceedings of the 10_{th} International Conference on World Wide Web.*, págs. 406–414. ACM (2001)

Freeman, L. Centrality in Networks: I. Conceptual Clarifications. Social Networks (1979)

Freeman, L.C. A set of measures of centrality based on betweenness. *Sociometry* págs. 35–41 (1977)

Freeman, L.C., Borgatti, S.P., y White, D.R. Centrality in valued graphs: A measure of betweenness based on network flow. *Social networks* **13**(2):141–154 (1991)

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., y Liu, T. Learning semantic hierarchies via word embeddings. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, tomo 1, págs. 1199–1209 (2014)

Ganesan, K., Zhai, C., y Viegas, E. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. En *Proceedings of the 21st international conference on World Wide Web*, págs. 869–878. ACM (2012)

Garimella, A., Banea, C., y Mihalcea, R. Demographic-aware word associations. En *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, págs. 2285–2295. (2017)

Geeraerts, D. 2.2 Meaning and definition. *A practical guide to lexicography* **6**:83 (2003)

Ghosh, U., Jain, S., y Soma, P. A Two-Stage Approach for Computing Associative Responses to a Set of Stimulus Words. En *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex-IV). COLING 2014 25th International Conference on Computational Linguistics*, págs. 15–21 (2014)

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., y Smith, N.A. Part-of-speech tagging for twitter: Annotation, features, and experiments. Informe técnico, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science (2010)

Gómez-Adorno, H., Bel-Enguix, G., Sierra, G., Sánchez, O., y Quezada, D. A machine learning approach for detecting aggressive tweets in spanish. En *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), CEUR WS Proceedings* (2018)

Gómez-Adorno, H., Reyes-Magaña, J., Bel-Enguix, G., y Sierra, G. Spanish Word Embeddings Learned on Word Association Norms. En *13th Alberto Mendelzon International Worshop on Fundations of Data Managment* (2019)

González, S.A. Índices de interés psicolingüístico de 1917 palabras castellanas. *Cognitiva* **8**(1):43–88 (1996)

Grover, A. y Leskovec, J. Node2vec: Scalable Feature Learning for Networks. En *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining.*, págs. 855–864. ACM (2016)

Guthrie, D., Allison, B., Liu, W., Guthrie, L., y Wilks, Y. A closer look at skip-gram modelling. En *LREC*, págs. 1222–1225 (2006)

Gómez-Adorno, H., Sidorov, G., Pinto, D., Vilariño, D., y Gelbukh, A. Automatic Authorship Detection Using Textual Patterns Extracted from Integrated Syntactic Graphs. *Sensors* **16**(9) (2016)

Hagberg, A., Schult, D., y Swart, P. Networkx: Python software for the analysis of networks. *Mathematical Modeling and Analysis, Los Alamos National Laboratory* (2005)

Halawi, G., Dror, G., Gabrilovich, E., y Koren, Y. Large-scale learning of word relatedness with constraints. En *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, págs. 1406–1414. ACM, New York, NY, USA (2012)

Hartmann, R.R. Lexicography and its interdisciplinary contacts, with special reference to linguistics and onomasiology. *Lexikos* **15** (2005)

Hassan, S. y Mihalcea, R. Cross-lingual semantic relatedness using encyclopedic knowledge. En *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3.*, págs. 1192–1201. Association for Computational Linguistics. (2009)

Hernández, L. *Creación semi-automática de la base de datos y mejora del motor de búsqueda de un diccionario onomasiológico.* Universidad Nacional Autónoma de México. (2012)

Hill, C. Alternatives to dictionaries. *Dictionaries, lexicography and language learning* págs. 115–121 (1985)

Hill, F., Reichart, R., y Korhonen, A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* **41**(4):665–695 (2015)

Hill, F., Cho, K., Korhonen, A., y Bengio, Y. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics* **4**:17–30 (2016)

Hochreiter, S. y Schmidhuber, J. Long short-term memory. *Neural computation* **9**(8):1735–1780 (1997)

Iezzi, D.F. Centrality measures for text clustering. *Communications in Statistics-Theory and Methods* **41**(16-17):3179–3197 (2012)

Jackson-Maldonado, D., Thal, D.J., Marchman, V.A., Newton, T., Fenson, L., y Conboy, B.T. *McArthur. Inventarios del desarrollo de habilidades comunicativas*. Brookes, Baltimore (2003)

Jarema, G., Libben, G., y Kehayia, E. The mental lexicon. *Brain and Language* **81** (2002)

Kahn, J.E. *Reader's Digest Reverse Dictionary: From the Word You Know to the Word You Need; how to Find the Words on the Tip of Your Tongue; Elusive Words, Awkward Words, Impressive Words, Persuasive Words, Precise Words, Technical Words* (1989)

Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **18**:39–43 (1953)

Kiss, G., Armstrong, C., Milroy, R., y Piper, J. *An associative thesaurus of English and its computer analysis*. Edinburgh University Press, Edinburgh. (1973)

Lafourcade, M. Making people play for Lexical Acquisition with the JeuxDeMots prototype. En *Proceedings of the 7th International Symposium on Natural Language Processing*, págs. 13–15. Pattaya, Thailand (2007)

Lancaster, F.W. *Information retrieval systems: characteristics, testing, and evaluation*. Information sciences series, 2ª edición. Wiley (1979)

Layton, R. *Learning data mining with python*. Packt Publishing Ltd (2015)

Leech, G. y Wilson, A. *Standards for Tagsets*, págs. 55–80. Springer Netherlands, Dordrecht (1999)

Levelt, W. Spoken word production: A theory of lexical access. *PNAS* **98**(23):13464–13471 (2005)

Liddy, E.D. Natural Language Processing. En *Encyclopedia of Library and Information Science*, 2ª edición. Marcel Decker, Inc., NY (2001)

Macizo, P., Gómez-Ariza, C.J., y Bajo, M.T. Associative norms of 58 Spanish words for children from 8 to 13 years old. *Psicológica* **21**:287–300. (2000)

Manning, C., Raghavan, P., y Schutze, H. *Introduction to Information Retrieval.* Cambridge University Press (2009)

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., y Zamparelli, R. A SICK cure for the evaluation of compositional distributional semantic models. En *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, págs. 216–223. European Language Resources Association (ELRA), Reykjavik, Iceland (2014)

Marmanis, H. y Babenko, D. *Algorithms of the intelligent web.* Manning Greenwich (2009)

McEvoy, C.L. Automatic and strategic processes in picture naming. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **14**(4):618 (1988)

Merriam, W. *Webster's New Dictionary of Synonyms: A Dictionary of Discriminated Synonyms with Antonyms and Analogous and Contrasted Words.* Merriam Webster (1984)

Mijangos, V., Barrón-Martinez, J.B., Arias-Trejo, N., y Bel-Enguix, G. A Graph-based Analysis of the Corpus of Word Association Norms for Mexican Spanish. En *2nd International Conference on Complexity, Future Information Systems and Risk*, págs. 87–93 (2017)

Mikolov, T., Chen, K., Corrado, G., y Dean, J. Efficient estimation of word representations in vector space. *Computing Research Repository.* **arXiv:1301.3781.** (2013)

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., y Joulin, A. Advances in Pre-Training Distributed Word Representations. En *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC'18 (2018)

Miller, G.A.e.a. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography* **3**(4):235–244 (1990)

Miller, G. y Charlees, W. Contextual correlates of semantic similarity. *Language and cognitive processes* **6**(1):1–28 (1991)

Nelson, D.L. y McEvoy, C.L. Encoding context and set size. *Journal of Experimental Psychology: Human Learning and Memory* **5**(3):292 (1979)

Nelson, D.L. y Schreiber, T.A. Word concreteness and word structure as independent determinants of recall. *Journal of memory and language* **31**(2):237–260 (1992)

Nelson, D.L., LaLomia, M.J., y Canas, J.J. Dissociative effects in different prime domains. *Memory & cognition* **19**(1):44–62 (1991)

Nelson, D.L., McEvoy, C.L., y Schreiber, T.A. *The University of South Florida word association, rhyme, and word fragment norms.* (1998)

Newman, M. *Networks: an introduction*. Oxford University Press (2010)

Newman, M.E. A measure of betweenness centrality based on random walks. *Social networks* **27**(1):39–54 (2005)

Nickels, L. When the words won't come: Relating impairments and models of speech production. *Aspects of Language Production* (2000)

Ogden, C.K. Basic English: A general introduction with rules and grammar (1930)

Padró, L. y Stanilovsky, E. FreeLing 3.0: Towards Wider Multilinguality. En *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*. ELRA, Istanbul, Turkey (2012)

Paivio, A., Yuille, J.C., y Madigan, S.A. Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of experimental psychology* **76**(1, Pt. 2):1–25 (1968)

Palermo, D.S. y Jenkins, J.J. Word association norms: Grade school through college. (1964)

Pennington, J., Socher, R., y Manning, C. Glove: Global vectors for word representation. En *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, págs. 1532–1543 (2014)

Pillai, S.U., Suel, T., y Cha, S. The Perron-Frobenius theorem: some of its applications. *IEEE Signal Processing Magazine* **22**(2):62–75 (2005)

Pinto, D., Gómez-Adorno, H., Vilarino, D., y Singh, V.K. A graph-based multi-level linguistic representation for document understanding. *Pattern recognition letters* **41**:93–102 (2014)

Pollux, J. *Onomasticon*, tomo 2 (1706)

Qi, F., Zhang, L., Yang, Y., Liu, Z., y Sun, M. WantWords: An Open-source Online Reverse Dictionary System. En *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, págs. 175–181. Association for Computational Linguistics, Online (2020)

Radinsky, K., Agichtein, E., Gabrilovich, E., y Markovitch, S. A word at a time: computing word relatedness using temporal semantic analysis. En *Proceedings of the 20th international conference on World wide web*, págs. 337–346. ACM, New York, NY, USA (2011)

RAE. *Diccionario de la Lengua Española.* Real Academia Española de la Lengua (2013)

Ratnaparkhi, A. A simple introduction to maximum entropy models for natural language processing. *IRCS Technical Reports Series* pág. 81 (1997)

Reyes-Magaña, J., Bel-Enguix, G., Gómez-Adorno, H., y Sierra, G. A Lexical Search Model Based on Word Association Norms. *Journal of Intelligent & Fuzzy Systems* **36**(5):4587–4597 (2019a)

Reyes-Magaña, J., Bel-Enguix, G., Sierra, G., y Gómez-Adorno, H. Designing an electronic reverse dictionary based on two word association norms of English language. *Electronic lexicography in the 21st century (eLex 2019): Smart lexicography* pág. 142 (2019b)

Reyes-Magaña, J., Sierra, G., Bel-Enguix, G., y Gómez-Adorno, H. Automatic Word Association Norms (AWAN). En *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, págs. 142–153 (2020)

Reyes-Magaña, J., Gómez-Adorno, H., Bel-Enguix, G., y Sierra, G. Representaciones vectoriales de palabras de un corpus de normas de asociación. *Research in Computing Science* **147**:109–118 (2018)

Robertson, S. y Zaragoza, H. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* **3**(4):333–389 (2009)

Robertson, S. y Sparck Jones, K. Relevance weighting of search terms. *Journal of the American Society for Information Science* **27**(3) (1976)

Roget, R. *Roget's Thesaurus of English Words and Phrases* (1911)

Roth, M. y Im Walde, S.S. Corpus Co-Occurrence, Dictionary and Wikipedia Entries as Resources for Semantic Relatedness Information. En *LREC* (2008)

Rubenstein, H. y Goodenough, J.B. Contextual correlates of synonymy. *Communications of the ACM* **8**(10):627–633 (1965)

Ruiz, P., Apud, I., Maiche, A., González, H., Pires, A.C., Carboni, A., Barg Beltrame, G., Martín, A., Aguirre, R., Moreira, K. *et al.* Manual de introducción a la psicología cognitiva (2016)

Sahlgren, M. The Distributional Hypothesis. *Italian Journal of Disability Studies* **20**:33–53 (2008)

Sanchez-Perez, M.A., Markov, I., Gómez-Adorno, H., y Sidorov, G. Comparison of Character N-grams and Lexical Features on Author, Gender, and Language Variety Identification on the Same Spanish News Corpus. En *International Conference of the Cross-Language Evaluation Forum for European Languages*, págs. 145–151. Springer (2017)

Sanfeliu, M.C. y Fernandez, A. A set of 254 Snodgrass-Vanderwart pictures standardized for Spanish: Norms for name agreement, image agreement, familiarity, and visual complexity. *Behavior Research Methods, Instruments, & Computers* **28**(4):537–555 (1996)

Santus, E., Chersoni, E., Lenci, A., Huang, C.R., y Blache, P. Testing APSyn against vector cosine on similarity estimation. En *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, págs. 1861–1870 (2016)

Sapkota, U., Bethard, S., Montes, M., y Solorio, T. Not all character n-grams are created equal: A study in authorship attribution. En *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, págs. 93–102 (2015)

Scholkopf, B. y Smola, A.J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press (2001)

Shcherba, L. Towards a General Theory of Lexicography. *International Journal of Lexicography* **8**(4):314–350 (1995)

Sidorov, G. Problemas actuales de lingüística computacional. *Revista digital universitaria, UNAM, México* **2**(1) (2001)

Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., y Chanona-Hernández, L. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications* **41**(3):853–860 (2014)

Sierra, G. Design of an onomasiological search sistem: A concept-oriented tool for terminology. *Terminology* **6**(1):1–34 (2000a)

Sierra, G. The onomasiological dictionary: a gap in lexicography. En *Proceedings of the Ninth Euralex International Congress*, págs. 223–235 (2000b)

Sierra, G. y McNaught, J. Extracting semantic clusters from MRDs for an onomasiological search dictionary. *International Journal of Lexicography* **13**(4):264–286 (2000)

Sierra, G. y McNaught, J. Natural Language System for Terminological Information Retrieval. En A. Gelbukh (editor), *Computational Linguistics and Intelligent Text Processing*, págs. 541–552. Springer, Berlin, Heidelberg (2003)

Sinopalnikova, A. y Smrz, P. Word Association Thesaurus as a Resource for extending Semantic Networks. En *Communications in Computing.*, págs. 267–273. (2004)

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., y Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. En *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, págs. 1631–1642 (2013)

Stephenson, K. y Zelen, M. Rethinking centrality: Methods and examples. *Social networks* **11**(1):1–37 (1989)

Thorat, S. y Choudhari, V. Implementing a Reverse Dictionary, based on word definitions, using a Node-Graph Architecture. *arXiv preprint arXiv:1606.00025* (2016)

Thorndike, E.L. y Lorge, I. The teacher's word book of 30,000 words. (1944)

Toglia, M.P. y Battig, W.F. *Handbook of semantic word norms.* Lawrence Erlbaum (1978)

Ullman, S. Semantics An Introduction to The Science of Meaning Oxford (1983)

Vilariño, D., Pinto, D., Gómez, H., León, S., y Castillo, E. Lexical-syntactic and graph-based features for authorship verification. En *Proceedings of CLEF*, págs. 282–302 (2013)

Wasserman, S. y Faust, K. *Social network analysis: methods and applications*. Structural analysis in the social sciences. Cambridge University Press (1994)

Widdows, D. y Dorow, B. A graph model for unsupervised lexical acquisition. En *19th International Conference on Computational Linguistics* (2002)

Witten, I.H. Text Mining. (2004)

Wright, D. Using word n-grams to identify authors and idiolects. *International journal of corpus linguistics* **22**(2):212–241 (2017)

Wuster, E. The machine tool: an interlingual dictionary of basic concepts comprising an alphabetical dictionary and a classified vocabulary with definitions and illustrations (1968)

Xie, Z. Centrality measures in text mining: prediction of noun phrases that appear in abstracts. En *Proceedings of the ACL student research workshop*, págs. 103–108. Association for Computational Linguistics (2005)

Zhang, D., Xu, H., Su, Z., y Xu, Y. Chinese comments sentiment classification based on word2vec and SVMperf. *Expert Systems with Applications* **42**(4):1857–1863 (2015)

Zheng, L., Qi, F., Liu, Z., Wang, Y., Liu, Q., y Sun, M. Multi-channel reverse dictionary model. En *Proceedings of the AAAI Conference on Artificial Intelligence*, tomo 34, págs. 312–319 (2020)

Zock, M. y Biemann, C. Comparison of Different Lexical Resources With Respect to the Tip-of-the-Tongue Problem. *Journal of Cognitive Science* **21**(2):193–252 (2020)

Zock, M., Schwab, D., y Rakotonanahary, N. Lexical Access, a Search-Problem. En *Proceedings of the 2nd Workshop on Cognitive Aspects of the Lexicon (CogALex 2010)*, págs. 75–84 (2010)

Zuhair, Mohamed, T., Al, K., y Seifedine. *Python for Graph and Network Analysis*. 1ª edición. Springer International Publishing (2017)

Apéndice A

Corpus de definiciones

A.1. Español

Tabla A-1: Definiciones correspondientes al campo semántico de medios de transporte

Autobús	**Avión**	**Barco**	**Bicicleta**	**Camión**
Transporte terrestre público de pasajeros muy grande	Medio de transporte con el que puedes volar	Un medio de transporte que navega por el mar	Transporte que no necesita de gasolina o derivados, se mueve gracias a la fuerza que realiza el humano. existen algunas estacionarias que son usadas para hacer ejercicio.	Transporte público
Transporte público grande	Un sistema de transporte que se va volando por el cielo.	Donde van las personas en el mar	La que pedaleas	Bus
Carro grande para llevar pasajeros que no es una combi o un metro	Medio de transporte que vuela	Zarpar	Es un transporte que se promueve los fines de semana en la ciudad	Medio de transporte grande con cuatro ruedas, que se utiliza para carga pesada
Transporte para muchas personas	Transporte aéreo	Balsa grande que flota en el mar	Vehículo sin motor y con dos ruedas	El transporte público.
Transporte público de pasajeros	Medio de transporte en el aire	Titanic	La moto sin motor	Medio de transporte público

Tabla A-2: Definiciones correspondientes al campo semántico de animales pequeños

Abeja	**Ardilla**	**Rana**	**Mariposa**	**Ratón**
Insecto volador rayado que produce miel	Roedor comedor de nueces	Un insecto de colores vivos que vuela	Es verde y húmedo, brinca y es un tipo de anfibio	Animal de color gris que puede transmitir la rabia. en conjunto es considerado como plaga. se reproducen de manera abismal.
Obreras pequeñitas	Nueces	Es un anfibio que salta muy alto. es verde y vive en lugares húmedos.	Esto en lo que se convierten las orugas cuando les salen alas	El animal con cola que anda en coladeras
Insecto que pica, los que hacen la miel	Animal que come nueces, vive y trepa en los árboles. cuatro patas, con pelaje café o grisaseo	Anfibio de color verde que salta y croa	Alas	Pequeño y le gusta el queso
Animal que produce miel	El animalito que come nueces. el que sale en la era de hielo.	Anfibio que salta	En lo que se convierten las orugas	Roedor perseguido por gatos
Insecto volador amarillo y negro	Animal roedor que come bellotas	Animalito que vive en el agua, es verde y hace croac	Monarca	El animal como mickey mouse

Tabla A-3: Definiciones correspondientes al campo semántico de mamíferos

Caballo	**Cebra**	**Cochino**	**León**	**Borrego**
Animal que se puede montar	El animal que parece caballo y tiene rayas blancas y negras	Animal consumido en méxico en carnitas, chicharrón y de él proviene la manteca.	Ruge y vive en la selva	Animal de granja con lana y que usan para la barbacoa
Un animal de cuatro patas que corre mucho y se usa para carreras.	Un animal que parece caballo pero con rayas	Con el que se hacen las carnitas	Rey	Animal con lana que no es alpaca
Animal equino usado para cabalgar, se usa en los juegos de polo	Equino	Que esta sucio	Animal carnívoro, de cuatro patas, grande melena, pelaje amarillo. es el rey de la selva	Animal acolchonado blanco
Animal que galopa y relincha	El animal con rayas blancas y negras	Parecido al cerdo o puerco	El animal del escudo de gryffindor.	Animal que produce lana
Animal que relincha	Burro	El animal de la alcancía	Animal conocido como el rey de la selva	Animal de un ganado que produce lana

Tabla A-4: Definiciones correspondientes al campo semántico de aves

Gallina	**Ganso**	**Guajolote**	**Pájaro**	**Búho**
El animal que pone los huevos	Es un ave muy parecida al pato pero más grande y, además, agresivo	Animal mexicano que es similar al pavo y se cocina también con mole	Vuela en el cielo y no es avión	Pájaro nocturno que caza ratones que tiene grandes ojos, que ulula y que usan para simbolizar la inteligencia y que no es la lechuza
Animal que pone huevos.	Un animal que parece pato pero no es un pato	El ave que es el pavo en estados unidos	Ave	Ave que voltea la cabeza
Ave que pone huevos y cacarea	Cisne	Se acompaña con mole	Ave de tamaño pequeño de diversos tipos y colores	Animal, con alas, ojos bonitos, que se parece al logo de sanborns
Ave que da huevos	Esos que parecen patos pero más grandes	Ave similar a una gallina grande	El animalito que vuela. del tipo que son las palomas y las urracas.	Ave de ojos grandes que simboliza inteligencia
Ave que pone huevos	Pato pato	El animal que dice “gordogordogordo”	Animal que vuela	Ave nocturna, refiere sabiduría

Tabla A-5: Definiciones correspondientes al campo semántico de prendas de vestir

Abrigo	**Camisa**	**Falda**	**Pantalón**	**Playera**
Ropa como suéter que se usa para el frío pero más largo	Tipo de playera que se usa para eventos formales, ir a trabajar y tiene cuello y botones	La prenda de ropa que usan, a veces, las mujeres en vez del pantalón	Ropa de vestir que generalmente está elaborado con mezclilla aunque puede ser hecho con otros materiales y es utilizado en su mayoría por hombres	Ropa deportiva
Lo que pones encima del suéter	Es una ropa que se usa para un evento formal, trabajo o estudio.	Es como un short pero sin lo de en medio de las piernas	La prenda de ropa que es de mezclilla	Ropa
Lo que usas para que ni te de frío	Prenda de vestir masculina que tiene botones al frente	Ropa	Prenda de vestir que se ajusta en la cintura y en las piernas	Prenda de vestir que usan tanto hombres como mujeres para cubrir la partes de arriba del cuerpo
Ropa que sirve para protegerse del frío	Vestimenta formal para hombres	Tela en forma circular para mujeres	Ropa que cubre de manera independiente las dos piernas y llega a los tobillos.	Lo que te pones encima, aquí *señalaría el torso*
Vestido que te cubre cuando hay frío	Es una prende vestir que tiene mangas y cuello	Ropa mujer	Prende de vestir para cubrir las piernas.	Prenda de vestir para la parte superior del cuerpo.

Tabla A-6: Definiciones correspondientes al campo semántico de electrodomésticos

Aspiradora	**Lámpara**	**Lavadora**	**Plancha**	**Radio**
Electrodoméstico que sirve para juntar polvo	Sirve para iluminar un espacio, lleva un foco	El aparato doméstico para lavar ropa	Instrumento doméstico que se utiliza para quitar las arrugar a la ropa. generalmente se vale del vapor, por lo tanto necesita agua para que funcione.	Escuchas música en el carro
Escoba eléctrica	Objeto que se necesita para que haya luz de noche.	Artefacto donde metes a lavar/ limpiar tu ropa	La herramienta que ocupas para desarrugar la ropa	Programa
Aparato para sacar el polvo	Aparato eléctrico usado para alumbrar lugares	Lavar	Se usa para las arrugas	Aparato electrónico en el que se puede escuchar música y sintonizar diferentes frecuencias
Aparato electrónico para quitar el polvo	Objeto que da luz por la noche	Donde se lava la ropa	Electrodoméstico para quitar arrugas de la ropa a base de calor	El aparato donde escuchas música. donde sintonizas estaciones.
Es un aparato para limpiar y succionar el polvo	Objeto que ilumina algún cuarto	Ropa	El aparato con el que la ropa queda lisa	Electrónico que sirve para escuchar programas y música.

Tabla A-7: Definiciones correspondientes al campo semántico de verduras

Calabaza	**Ejote**	**Elote**	**Papa**	**Zanahoria**
Verdura amarilla o naranja muy grande y redonda que usan para hacer lámparas	Verdura verde larga parecida a los chícharos	Base para los esquites	Es un tubérculo que puede ser comida chatarra, como las sabritas	Vegetal de color naranja, crece bajo la tierra y es el alimento preferido por los conejos, caballos, al menos en el imaginario colectivo.
Adornos de halloween	Unas verduras de la cual se prepara con huevo, es larga y verde.	Tortilla	Verdura blanquita que crece abajo de la tierra	la verdura que comen los conejos, ¿cómo se dice?
La cosa naranja a la que le ponen cara en halloween	Verdura de color verde que se compone de una vaina	Fruto que crece de una milpa, con granos, color como amarillo y blanco	Tubérculo	Comerla es buena para la vista
Verdura de color verde y amarilla por dentro, de sabor dulzón	Verdura larga pequeña verde	De donde vienen los esquites. que le echan mayonesa o lo asan.	Frituras	Raíz naranja que comen los conejos de las caricaturas
Verdura verde con pepita para hacer consomé	Es una vaina verde	Alimento típico para la elaboración de comida mexicana.	A la francesa	Lo que comen los conejos

Tabla A-8: Definiciones correspondientes al campo semántico de postres

Chocolate	**Dulce**	**Galleta**	**Gelatina**	**Helado**
Alimento dulce y café de origen prehispánico que a casi todos les gusta y produce endorfina.	Alimento con mucha azúcar que es rico	La tomas con leche, por lo regular son redondas	Postre elaborado con agua, puede ser de diversos sabores. este postre es recomendado para que los enfermos lo consuman.	Napolitano, frío, cono
Golosina	Esa sensación que produce comer azúcar.	Postre que se hornea y es planito y crujiente y puede tener chispas de chocolate	La que se hace con agua y cuaja	Frío
Sabe dulce, es de color café y viene del cacao	Sabor característico de los alimentos con azúcar	Hornear	Sus sabores son de frutas y nunca faltan en los cumpleaños	Postre formado por un cono de galleta y bolas de una mezcla fría de diferentes sabores
Postre de sabor dulce y preshispánico elaborado con cacao	Caramelo que se le da a los niños	Masa crujiente en forma redonda	Postre de sabores a base de agua que se deja cuajar	Postre. el del conito.
Bebida caliente muy dulce de méxico	Golosinas le gusta mucho a los niños	Polvorón	El postre viscoso que dan en las fiestas	Alimento dulce, postre frío.

Tabla A-9: Definiciones correspondientes al campo semántico de frutas

Durazno	**Fresa**	**Manzana**	**Melón**	**Plátano**
Fruta dulce con hueso no muy grande y con vellitos blancos	Fruta pequeña que es roja con un rabito verde, se hace mermelada con ella	La fruta que es redonda y puede ser de color rojo, verde y amarillo	Fruta color naranja pastel usualmente se encuentra en los cócteles de fruta. tiene cáscara dura, es necesario utilizar un cuchillo para poder consumirla.	Fruta amarilla y fálica
Color como el melocotón pero más naranja	Una fruta roja se hace mermelada con ella.	Una fruta roja que le das a tu maestra	La fruta que es rosa por dentro, que también es un color.	Mono
Fruta, suave, cáscara roja o amarilla, sabor dulce	Única fruta que tiene sus semillas exteriores y es de color rojo, es pequeña y de sabor dulce.	Fruta	Fruta anaranjada que es china	Fruta con cáscara color amarilla, por dentro de un color más blanco y espeso. la comen los monos :p
Fruta color naranja de sabor dulce y aterciopelada	Fruta dulce pequeña, roja	La fruta a la que le ponen chamoy y miguelito	Fruta redonda de cáscara clara, rugosa e interior naranja	Lo que comen los changos.
Fruto con semilla grande, aterciopelado	Fruta roja chiquita con semillas	Blanca nieves	La fruta redonda y grande... con semillitas	Fruta amarilla que es conocida por se alimento de monos

Tabla A-10: Definiciones correspondientes al campo semántico de alimentos

Espagueti	**Hamburguesa**	**Pan**	**Queso**	**Taco**
Alimento de harina italiano en fideo que va con crema o en jitomate	Comida rica que lleva carne entre dos panes, lechuga, jitomate, queso y catsup	El alimento que no puede faltar en el desayuno y que engorda	Alimento elaborado con leche. existen diferentes tipos: manchego, cotija. panela entre otros.	Comida mexicana, tortilla
Pasta que se enreda	Tiene carne, lechuga, tomate con tapa y base de pan.	Esa comida esponjocita que se hace con harina y se hornea, y se puede comer con leche o café	El producto que se saca de la leche de la vaca	Típico
Como sopa de fideo, pero seca	Alimento compuesto de dos panes y carne en la parte central, lleva lechuga, jitomate, pepinillos y cebolla.	Cebada	Amarillo y con agujeros	Platillo gastronómico elaborado de tortilla y puede ser de diferentes tipo de carne, longaniza y otros ingredientes
Comida típica de Italia elaborada de pasta con forma tubular y alargada	Comida estadounidense que tiene pan y carne	Bolillo	Derivado lácteo que ponen en trampas para ratones	Lo que haces rollito con la tortilla.
Ingrediente básico de la alimentación italiana	Comida chatarra que se sirve con pan	Concha	Como la crema pero sólido	Todo lo que cabe en una tortilla

Tabla A-11: Definiciones correspondientes al campo semántico de partes del cuerpo

Boca	**Brazo**	**Mano**	**Ojo**	**Pie**
Agujero con dientes y lengua que usamos para hablar y para comer	Parte del cuerpo en donde están las manos	La parte del cuerpo con la que agarras los objetos y que tiene 5 dedos	Parte del cuerpo que se utiliza para ver	En el zapato
Con lo que besas	Una parte del cuerpo, que sostiene cosas y aplica fuerza.	La parte del cuerpo conde terminan los brazos	*señalo la parte del cuerpo y pregunto, ¿cómo se llama?	Caminar
Lo que usas para hablar y comer	Extremidad superior del cuerpo humano	Extremidad	Tenemos dos y pueden ser de diferentes colores	Parte del cuerpo con la que caminamos y podemos sentir el suelo
Parte del cuerpo humano con la que comemos	Parte del cuerpo que te permite agarrar cosas	Con lo que agarran las cosas los humanos	Órgano que usamos para ver	Lo que está al final de la pierna. lo que te sostiene.
Parte del cuerpo, tiene labios y te sirve para hablar	Es una parte del cuerpo pegada a la mano	Pie	Con lo que vemos	Extremidad inferior

A.2. Inglés

Tabla A-12: Definiciones en inglés para banca, cubeta y ropa

Bench	***Bucket***	***Clothes***
You can sit on it in the street or a park and they are made of wood	*It's used for carrying water or other liquids. Also things like sand. A beach it is often with a spade and is used to make sandcastles*	*It describes the set of things you wear like trousers, jumpers, shirts, dresses, etc., anything in fact that you wear*
A long hard seat for several persons on which the players on a sport team sit	*A vessel for carrying liquid or solids. Has a handle. Often seen in wells and accompanies a spade when see on the beach*	*The collective name for the items which we wear; i.e. trousers, shirts, jumpers, etc.*
An object for sitting on, usually long which can seat many people sit on it (a few people can) in parks, made of wood or iron	*Item used to carry certain objects - mainly sand, water or soil. Children use them to make sandcastles at the beach*	*Buy it in shops, keeps us warm, can be fashionable or unfashionable. Consists of things such as jumpers, trousers - the collective name*
Object used for sitting on. Often found in public places such as parks and gardens. Used to seat 1 or more people at a time	*Used to carry water in/make sandcastles; has a handle*	*Things used to cover up and keep warm, used by humans*
Something you seat on, is longer than a chair, usually made of wood	*Used to collect/hold water, also on seaside to measure sand*	*Items people can wear*
Long platform for sitting on (fit many people on one)	*A device made out of metal or plastic used mainly to carry water or other fluids*	*Items that we wear, to keep us warm*
Apparatus for sitting on, designed for more than one person, often found in parks	*When cleaning the floor you put water in this and dip the mop into it*	*Garments worn on body*
A kind of seat found in parks, made of wood	*A device usually used to contain water*	*What you wear on your body*
A type of chair	*Item with a handle used for carrying things in, usually made of plastic*	*Items that we all wear*
Often found in public places such as parks and gardens. Used to seat 1 or more people at a time	*A sort of container used to carry something*	*Things you wear*

Tabla A-13: Definiciones en inglés para huracán, limón, ardilla y agua

Hurricane	***Lemon***	***Squirrel***	***Water***
A kind of severe storm which involves a lot of strong winds	*It's a yellow fruit, like limes. Citrus. Used in cooking for sharpness*	*It's a little rodent and can be red or grey, it has a big bushy tail*	*It's a clear liquid that you get from a tap*
A violent tropical cyclone	*A yellow citrus fruit. Sour tasting. Often used as an accompaniment to drinks*	*A small rodent living in trees with a long bushy tail*	*The colourless transparent liquid occurring on rivers*
Very strong winds of over 100 mph which cause great destruction	*a yellow citrus fruit with a bitter taste often sliced and put in drinks*	*A small rodent which lives in trees, collects nuts and has a bushy tail*	*A clear, neutral liquid that surrounds us everywhere*
Strong winds and rain, gale, destruction over large areas	*It's a citrus fruit, yellow, used with sugar on pancakes*	*Animal, grey/red, bushy tail, lives in trees, buries nuts*	*Liquid, clear, drinkable – constituents are hydrogen and oxygen*
A very strong wind, very destructive, often named after people's first names, i.e. the weather researchers who first discover each one	*It's a yellow citrus fruit. Tastes bitter. Oval shaped*	*Small animal, lives in trees, eats acorns, has a bushy tail*	*Liquid, clear, H2O*
Big amount of wind going from country to country, ruining everything as it goes!	*A yellow sour fruit*	*Animal, bushy tail, eats nuts, builds nests in trees called dreys*	*Liquid form, scientific term H2O*
Violent winds which can cause large scale destruction	*A yellow citrus fruit*	*Small funny animal with big, bushy tail, likes nuts, likes trees*	*Liquid, freezes at 0°C*
Natural disaster, whipping wind, destroys anything in its path	*Yellow, citrus, fruit*	*Animal that lives in trees and collects acorns, has a long tail*	*Liquid, clear, boils at 100°C, freezes at 0°C*
A type of typhoon	*Citrus fruit which is yellow*	*A small-sized animal, habitat in trees*	*Fluid, clear, tasteless, colourless*
A very very strong wind	*Yellow citrus fruit*	*Small grey mammal, relative to the rodent, found in both countryside and town*	*Wash with it; drink it; used for dilution; H2O; found in springs, rivers, lakes, seas, oceans*