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

VALIDACIÓN DE REPRESENTACIONES VECTORIALES DE PALABRAS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

THIBAULT BENOIT SWYSEN CACHAÑA

PROFESOR GUÍA:
JORGE PÉREZ ROJAS

MIEMBROS DE LA COMISIÓN:
BÁRBARA POBLETE LABRA
JUAN BARRIOS NUÑEZ

SANTIAGO DE CHILE
2020

Resumen

Los word embeddings, también denominados representaciones vectoriales de palabras, son vectores de números reales, de pocas dimensiones, los cuales son utilizados en la resolución de distintas tareas relacionadas al procesamiento de texto.

Parte importante del desarrollo de word embeddings, es determinar la capacidad de representación semántica y sintáctica que estos poseen. Para realizar esto, los word embeddings son evaluados utilizando distintos métodos. Estos métodos se dividen en dos tipos: métodos intrínsecos y métodos extrínsecos. Los métodos intrínsecos comparan la relación semántica entre palabras respecto a los vectores correspondientes a dichas palabras. En cambio, los métodos extrínsecos consisten en evaluar el desempeño de los word embeddings al resolver tareas relacionadas al procesamiento de texto.

Uno de los principales problemas con la validación de word embeddings, es el poco desarrollo que hay para lenguajes distintos al inglés, entre ellos el español. Producto de esto, no existen herramientas con las cuales evaluar word embeddings para el español.

Producto de lo anterior, en este trabajo de titulación, se creará una herramienta con la cual evaluar word embeddings para este lenguaje. A su vez, se busca evaluar una serie de word embeddings del lenguaje español

La herramienta de evaluación desarrollada permite la evaluación de word embeddings utilizando métodos intrínsecos y extrínsecos. Los métodos de validación intrínseca implementados fueron: similitud semántica, analogías de palabras, outlier detection y cross-match. Mientras que, para la validación extrínseca, se utilizó una tarea de clasificación de texto.

Una vez desarrollada la herramienta de evaluación, esta se utilizó en distintos modelos de word embeddings. A partir de los resultados obtenidos en las evaluaciones, se observó que los word embeddings generados a partir del algoritmo FastText en general obtuvieron los mejores resultados. Mientras que, los word embeddings generados por los algoritmos GloVe y Word2Vec, obtuvieron resultados mixtos en las distintas evaluaciones. Finalmente, el word embeddings BETO en general obtuvo los resultados más bajos, aunque obtiene resultados cercanos al resto de word embeddings durante la evaluación extrínseca.

Finalmente, se concluyó exitosamente la creación de una herramienta para la evaluación de word embeddings. Esta herramienta permite la comparación de distintos word embeddings, a través de diferentes métodos de evaluaciones.

Dedicado a la Carmen.

Agradecimientos

Nunca he sido bueno para expresarme con palabras, por lo que no escribiré mucho, pero eso no quiere decir que no le agradezca la gente que me ha apoyado en mi vida, sobre todo durante mi paso por la universidad. Una primera persona a la que quiero agradecer es a Jorge, quien me ayudó durante el desarrollo de este trabajo.

También quiero agradecer a mis amigos: Zoka, Elias, Rayo, la flaca, Baño, el Dani, Marinkovic, el Ale, el Jose, Cristopher, el pelao Seba, el Andres, quienes fueron los que me apoyaron durante mi paso por la universidad, y algunos, durante mucho más.

Finalmente, quiero agradecer a mi madre, quien me preparo un café todos mientras estuve en la universidad, mi padre, quien me ha acompañado a todos los partidos de tenis en los que jugué, a mis hermanos, por hacerme los días fueran más alegres, a mis abuelos, por siempre estar conmigo, y a mi familia, por ser mi familia.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Contexto

En el día a día, las personas interactúan con diversas tecnologías, las cuales son fruto de la investigación y el desarrollo en diversas áreas, siendo el área de la inteligencia artificial una de las que más se ha hecho notar en el último tiempo. Ejemplo de lo anterior, es el uso de técnicas de inteligencia artificial en aplicaciones como FaceApp (aplicación para alterar imágenes del rostro de una persona), en investigaciones científicas tales como la primera imagen que se obtuvo de un agujero negro, o en el desarrollo de vehículos autónomos.

Un área que pasa ligeramente desapercibida, dentro del ámbito de la inteligencia artificial, es la del procesamiento del lenguaje natural o, por sus siglas en inglés, NLP. En el área de NLP, se estudia la interacción entre computadores y el lenguaje humano (lenguaje natural) desde distintos puntos de vista, como la sintaxis o la semántica. Algunas de las tareas que son estudiadas en esta área son traducción automática (machine translation) y la generación de resúmenes de texto (automatic summarization), entre otras. Algunas de las aplicaciones del estudio de NLP más conocidas en la actualidad, son la creación de chatbots y de asistentes virtuales inteligentes.

Parte importante en NLP, es la representación que se le da a las palabras, no desde un punto de vista sintáctico, sino que desde un punto de vista semántico, pensando en cómo puede una máquina entender el significado que las personas le dan a las palabras. Para ello, se han desarrollado diversos métodos, los cuales intentan encapsular el significado de las palabras, como la utilización de redes semánticas como WordNet o la aplicación de Brown Clustering. Sin embargo, en la actualidad, una de las formas más utilizadas para representar el significado de las palabras es a través de vectores densos de palabras, más conocidos como *word embeddings*.

Los word embeddings, también denominados representaciones vectoriales de palabras, son vectores de números reales, de pocas dimensiones, los cuales son utilizados como parte de la resolución de distintas tareas de procesamiento del lenguaje natural. Esta forma de representar palabras como vectores se basa en la idea de que una palabra obtiene su significado a partir del contexto que le acompaña dentro de grandes volúmenes de texto, entendiéndose

el contexto como el resto de palabras que están acompañando en alguna oración. Algunos algoritmos a partir de los cuales se pueden generar estos vectores son Word2Vec [12], GloVe [15] y FastText [2].

Parte importante de la generación de word embeddings, es saber cuál es la capacidad de representación semántica que estos poseen. Determinar la capacidad representativa de los word embedding se logrará a partir del uso de diversos métodos de validación, los cuales se clasifican principalmente en métodos de validación intrínsecos y métodos de validación extrínsecos. Los métodos de validación intrínsecos comparan la relación semántica entre palabras respecto a los vectores correspondientes a dichas palabras. En cambio, los métodos de validación extrínsecos consisten en evaluar el desempeño de word embedding dentro de distintas tareas de NLP.

De la literatura, podemos observar que existe una gran cantidad de métodos para validar word embeddings, tanto de tipo intrínseco como extrínseco. Sin embargo, el enfoque principal ha sido para la validación de word embeddings para el lenguaje inglés, por lo cual, es difícil encontrar trabajos sobre validación de word embedding para otros idiomas como el español.

1.2. Alcance

En el presente trabajo se expone el desarrollo de una herramienta para la evaluación de word embeddings para el lenguaje español. La cual podrá ser utilizada para realizar cuatro tipos de evaluaciones intrínsecas y una evaluación extrínseca.

Además, se exponen los resultados de evaluaciones realizadas a distintos word embeddings específicos del lenguaje español, utilizando la herramienta de evaluacion desarrollada. Los resultados presentados corresponden a la evaluación de word embeddings generados a partir de distintos métodos, utilizando diferentes corpus de texto. Se espera que este trabajo entregue información útil, al momento de decidir qué tipo word embedding utilizar en tareas de NLP.

1.3. Objetivo general

El objetivo de este trabajo de titulación es el desarrollo de un programa que pueda ser utilizado como herramienta para la evaluación de distintos word embeddings para el español. Es importante destacar, que lo anterior es un aporte al desarrollo de word embeddings para el español, ya que no existe un conjunto de evaluaciones estandarizadas para este lenguaje.

Adicionalmente, un segundo objetivo es el de obtener resultados sobre la evaluación de distintos word embeddings específicamente computados para el lenguaje español, los cuales han sido generados a partir de distintos algoritmos y corpus de textos. Más en detalle, las evaluaciones realizadas deben ser tanto de tipo intrínseco como extrínseco.

1.4. Objetivos específicos

Para alcanzar el objetivo general propuesto, se necesitará cumplir con los siguiente puntos:

1. **Definir el conjunto de métodos de validación a implementar**. Es importante determinar qué métodos de validación pueden ser implementados, principalmente debido a las limitaciones de tiempo que existen para este trabajo. Estos métodos a implementar tienen que cumplir con que: 1) las métricas de evaluación a utilizar deben estar presentes en la literatura y ser factibles de implementar en un tiempo razonable, y 2) en caso de necesitar de un dataset específico, este tiene que ser posible de obtener también en un tiempo razonable.

2. **Implementación de los distintos métodos de validación seleccionados**. La implementación de los métodos de validación, incluye la implementación de las métricas a utilizar, además de establecer, de ser necesario, datasets para las evaluaciones.

3. **Desarrollar un programa con el cual controlar la evaluación de un word embedding**. Este programa debe permitir seleccionar cuáles métodos de validación utilizar al evaluar un word embedding.

1.5. Metodología

El desarrollo de este trabajo se realizó en distintas fases. La primera fase consistió en investigar qué métodos de validación existen en la literatura. Al finalizar esta fase de investigación, se terminó con un conjunto de métodos designados como candidatos iniciales a ser implementados. Los métodos seleccionados en esta fase, corresponden a aquellos que no requerían de un dataset para realizar evaluaciones o, de necesitar un dataset, se determinó que la obtención de este era posible.

Una vez definido un conjunto inicial de métodos de validación, se continuó con elegir aquellos que serían finalmente implementados. Durante la selección de los métodos a implementar, los principales aspectos evaluados fueron las métricas y datasets que se utilizarían. En esta fase, en caso de que fuese necesario, se realizó la búsqueda de datasets a utilizar para la implementación de la herramienta de evaluación (aunque, en un caso particular, se construyó un dataset propio). Al finalizar esta fase, se definieron los métodos de validación a implementar, además de los dataset y métricas a utilizar.

Siguiendo con la implementación de los métodos elegidos, esto fue realizado a través del lenguaje de programación Python, el cual fue elegido debido a su facilidad de uso y la gran cantidad de librerías que existe para realizar proyectos relacionados con inteligencia artificial.

Una vez implementados los distintos métodos de validación elegidos, se desarrolló un programa con el cual realizar la evaluación de word embeddings. Este programa permite controlar cuáles métodos de validación son utilizados al evaluar embeddings. Finalmente, se utilizó este programa para realizar la validación de distintos word embeddings del español, a través de los diferentes métodos de validación implementados.

1.6. Estructura del documento

El resto de este documento se estructura como sigue. El Capítulo 2 muestra cuál es el marco teórico y el estado del arte, definiendo que es un word embedding y describiendo los métodos con los que pueden ser obtenidos. En esta sección también se encuentra una descripción de los trabajos relacionados a la validación de word embeddings, además de una descripción de los distintos métodos de validación desarrollados, tanto de tipo intrínseco como extrínseco.

El Capítulo 3 describe aspectos generales de la implementación y funcionamiento de la herramienta de evaluación creada. A su vez, también se entrega una descripción de los dataset y métricas utilizadas en cada uno de los métodos implementados junto a la herramienta de evaluación. Capítulo 4 presenta los resultados de las evaluación de distintos word embeddings para el español, además de su correspondiente análisis. Finalmente, en el Capítulo 5 se presentan las conclusiones y el trabajo a futuro.

Capítulo 2

Marco Teórico y Estado del Arte

En este capítulo se expone información sobre word embeddings, así como algunos métodos con los cuales generarlos. También se mencionan distintos trabajos sobre métodos de evaluación para word embeddings, al igual que trabajos sobre la evaluación de word embeddings para el español. Finalmente, se presentan métodos intrínsecos y extrínsecos. Estos métodos corresponden a los implementados para el desarrollo de la herramienta de evaluación.

2.1. Word Embeddings

Los word embeddings, o representaciones vectoriales de palabras, son vectores de números reales, de pocas dimensiones, los cuales intentan representar el significado semántico y sintáctico de las palabras. La construcción de estas representaciones vectoriales se basa principalmente en la "hipótesis distribucional", la cual establece que las palabras que ocurren en el mismo "contexto" tienden a tener significados similares, o dicho de forma más conveniente, una palabra se caracteriza a partir de las palabras que le acompañan [17].

Existen diversas formas de generar estos word embeddings, por ejemplo, a través del uso de redes neuronales, matrices de co-ocurrencia, Brown clusters o modelos probabilísticos, siendo las redes neuronales el método más popular en la actualidad. Algunos métodos que se utilizan actualmente para la generación de word embeddings son:

- **Word2Vec**[24]: Utilizando una red neuronal de dos capas, existen dos métodos para generar word embeddings. El primer método consiste en entrenar la red neuronal para predecir el contexto más apropiado para una palabra dada (Skip-gram), mientras que, para el segundo método, la red es entrenada para predecir la palabra más probable para un contexto dado (CBOW). En ambos métodos, los word embeddings se extraen desde la primera capa de la red neuronal.

- **FastText**[4]: Algoritmo basado en Skip-gram, en el cual, durante el proceso de entrenamiento de la red neuronal, además de las palabras, también se utilizan los n-gramas[1] de dichas palabras.

[1]Un n-grama es una subsecuencia continua de tamaño n para una secuencia dada.

- **GloVe** [27]: Este algoritmo se basa en el uso de matrices de co-ocurrencia, las cuales representan la frecuencia de ocurrencia de palabras respecto a otras, donde el valor del elemento $x_{l,j}$ de la matriz de co-ocurrencia X, corresponde a la cantidad de veces que la palabra l aparece junto a la palabra j.

- **BERT (Bidirectional Encoder Representations from Transformers)** [11]: Al igual que algoritmos anteriores, BERT está basado en redes neuronales, con la diferencia de que utiliza un tipo de red neuronal especial, denominada Transformer. Por otra parte, el vocabulario de BERT no sólo contiene palabras, también contiene sub-palabras y caracteres individuales, esto es para poder representar palabras fuera del vocabulario. Utilizando las sub-palabras y caracteres, este modelo puede definir un vector representante para las palabras desconocidas. Cabe destacar que, si bien es posible generar word embeddings a partir de BERT, esta no es su función principal.

2.2. Trabajos relacionados

Son variados los trabajos relacionados a los métodos de validación para word embeddings (de tipo intrínseco y extrínseco). Buena parte de estos, tienen por tema principal presentar un algoritmo para generar word embeddings, y solo exponen métodos de validación al momento de evaluar dichos algoritmos. También existen trabajos que presentan y describen distintos métodos de validación existentes [1] [18]. Otros trabajos examinan las capacidades de evaluación que presentan algunos métodos [14] [32] [29], y también hay aquellos que examinan la relación entre métodos intrínsecos y extrínsecos de validación [34] [36].

Respecto a trabajos sobre la validación de word embeddings del lenguaje español, la cantidad de estos es reducida. Lo anterior puede deberse a que, al hablar de embeddings de lenguajes distintos al inglés, el enfoque actualmente está en el desarrollo de word embeddings multilinguals o cross-linguals, los cuales consisten en embeddings que trabajan con más de un lenguaje al mismo tiempo. De los trabajos encontrados, ambos consisten en el desarrollo y evaluación word embeddings generados a partir de un algoritmo específico. En [13], el embedding fue creado utilizando el algoritmo GloVe y evaluado sólo con métodos intrínsecos (similitud semántica y analogías de palabras), mientras que, en [33], se crea un embedding a partir del algoritmo FastText y evaluado con un método intrínseco (similitud semántica) y uno extrínseco (reconocimiento de entidades). Este último embedding, fue desarrollado de forma tal que interpreta las palabras con un enfoque hacia el área de la medicina.

2.3. Estado del Arte

Dentro de la literatura existen diversos métodos con los cuales evaluar la calidad representativa de word embeddings. Estos métodos se clasifican principalmente en dos grupos, aquellos métodos que evalúan los embeddings directamente frente a distintas tareas dentro de NLP, denominados métodos extrínsecos, y los métodos que comparan los embeddings frente al juicio humano o la interpretación que se tiene de las palabras, denominados métodos intrínsecos.

Si bien los word embeddings no son algo muy reciente, aún existen ciertos detalles sobre la validación de word embeddings los cuales no están del todo claros o que faltan por cumplir correctamente:

- **No hay claridad sobre qué cosas hacen un buen embedding.** De forma intuitiva, se considera que un word embedding es bueno, si este logra representar correctamente la relación que existe entre palabras, pero, no está completamente definido que tipo de relaciones deberían estar representadas.

- **Correlación entre validación intrínseca y extrínseca.** Los métodos de validación se dividen principalmente en dos tipos: métodos intrínsecos y métodos extrínsecos, sin embargo, no está del todo claro cómo se correlacionan los resultados de ambos métodos. Lo anterior implica que, tener un buen rendimiento con algún método intrínseco, no implica tener buenos resultados con métodos extrínsecos, y viceversa. Sobre este tema, existen varios trabajos que investigan si existe alguna correlación entre los resultados provenientes de distintos tipos de métodos [36] [34] [10] [28].

- **Hubness Problem.** Algunas palabras tienden a ocurrir de forma más frecuente en diversos tipos de contextos, esto se puede traducir en inconsistencias dentro de los embeddings producto de la aparición de "hubs", vectores que se encuentran muy cercanos a otros vectores dentro del embedding [14] [32]. La presencia de "hubs" dentro del embedding, puede influenciar los resultados obtenidos a través de similitud coseno, la cual es utilizada en varios métodos de validación, como una medida de similitud entre palabras.

- **Presencia de subjetividad en algunos métodos intrínsecos.** Como es mencionado en [36] y [14], en algunos métodos de validación se puede presentar cierta subjetividad. Ejemplos de lo anterior, son los test de similitud semántica y analogías, debido a que estos test dependen del razonamiento humano, por lo que pueden estar influenciado por diversos factores, como sociales y culturales.

- **Polisemia.** La existencia de palabras que pueden tener más de un solo significado, como por ejemplo la palabra "banco", la cual puede hacer referencia a la institución financiera o un asiento, presentan un desafió al momento de intentar representarlas en un embedding o al evaluar la calidad de estos mismos.

2.3.1. Validación intrínseca

Los métodos de validación intrínsecos evalúan la capacidad que tiene un embedding para representar la relación semántica o sintáctica que existe entre palabras. Para realizar esto, las relaciones entre palabras que aparecen dentro de los word embeddings son comparadas con las relaciones que, a juicio humano, existe entre las mismas.

Si bien los métodos de validación son separados en extrínsecos e intrínsecos, algunos trabajos distinguen distintas subclases dentro de los métodos de validación intrínsecos. Ejemplo de lo anterior es [32], en donde se separa los métodos de validación según dos criterios: aquellos métodos en los que se evalúan los embeddings de forma individual para luego compararlos

entre ellos (evaluación intrínseca absoluta) y aquellos en los que se les pregunta directamente a las personas cuál es el embedding más apropiado (evaluación intrínseca comparativa).

Adicionalmente, en [1] también se presenta una separación en subclases, a partir de diferenciar cuál es la procedencia de los datasets o elementos utilizados para realizar los test de validación:

- Evaluación Intrínseca Consciente: El dataset utilizado para validar un embedding es recolectado de forma off-line, es decir, las personas consultadas durante la construcción del dataset tienen tiempo para pensar su respuesta.

- Evaluación Intrínseca Subconsciente: El dataset utilizado para validar un embedding es recolectado de forma on-line, es decir, las personas consultadas durante la construcción del dataset deben responder de forma inmediata.

- Evaluación Intrínseca basado en Tesauros: Estos métodos se basan en la comparación de embeddings con bases de datos de conocimiento, redes semánticas y tesauros.

- Evaluación Intrínseca basado en el Lenguaje: Estos métodos se basan en la comparación de embeddings con elementos subyacentes en el lenguaje, como la representación grafemática de las palabras o la frecuencia de pares de palabras dentro de un corpus.

Por otro lado, [36] define una serie de características importantes a tener en cuenta al momento de definir o utilizar algún método de validación:

- Los dataset utilizados para realizar los test de validación deben abarcar una gran variedad de palabras.

- Debe existir correlación entre los resultados de los test realizados y el rendimiento en tareas de NLP.

- Los experimentos deben ser capaces de evaluar varias propiedades del modelo.

- Al realizar los experimentos, estos tienen que ser computacionalmente eficiente.

- Los resultados del test deben ser representativos, es decir, tener significancia estadística, al momento de comparar distintos embeddings.

Finalmente, los siguientes son algunos de los métodos intrínsecos de validación presentes en la literatura y su correspondiente descripción. Estos métodos son los que han sido implementados en el presente trabajo para la validación de word embeddings, una mayor cantidad de métodos intrínsecos es descrita en el trabajo de [1].

2.3.1.1. Similitud semántica

Uno de los métodos de validación intrínseca más utilizado dentro de la literatura, es la validación por similitud semántica, principalmente debido a lo sencillo y eficiente que es de realizar. La idea de este método es evaluar la capacidad representativa de los embeddings a través de medir qué tanto pueden encapsular la relación que existe entre pares de palabras. Concretamente, este método busca determinar cuál es el grado de correlación entre la "similitud" de pares de palabras, respecto a la "similitud" de los pares de vectores correspondientes en el embedding.

Para la realización de este método, se necesita de un conjunto de pares de palabras con los cuales realizar los experimentos, definir qué es la similitud entre pares de palabras y qué es la similitud entre vectores de un embedding. Los datasets utilizados para la realización de los experimentos, consisten en varios pares de palabras (en versiones más recientes se utilizan pares de conceptos, los cuales pueden consistir de más de dos palabras), a los que se les asigna un valor numérico, el cual corresponde al grado de similitud existente entre el par de palabras correspondiente (aunque hay casos donde la construcción del dataset no ocurre de la misma forma [6] [21]). Usualmente, este valor numérico se obtiene después de preguntar directamente a un grupo de personas sobre el valor de similitud entre palabras, una vez definida una escala (por ejemplo, cero implica palabras antónimas o sin ninguna relación, y cuatro implica palabras sinónimas). La Tabla 2.1 es un ejemplo del posible contenido dentro de los dataset.

Par de palabras		Similitud
PlayStation	Wii	3.17
soldado	paz	0.67
Apple	iPhone	2.5
fiscal	abogado	3.25
dióxido de carbono	CO2	4.0

Tabla 2.1: Ejemplo de contenido de un dataset de similitud semántica.

La similitud entre pares de vectores se define como una función de distancia, siendo la distancia coseno la más utilizada en la literatura. Esta distancia coseno, se define como el coseno del ángulo formado entre un par de vectores.

$$cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} \tag{2.1}$$

El resultado final del test se obtiene a partir de la correlación que hay entre los dos valores de similitud, de pares de palabra y pares de vectores. Como medida de correlación, es usual utilizar los coeficientes de correlación de Pearson (r), Spearman (ρ) o Kendall (τ). Estos coeficientes adoptan valores entre -1 y +1, donde un valor cercano a +1 implica que existe una relación creciente entre la similitud percibida en las palabras y la similitud presente en el embedding. En cambio, un valor de -1 implica que la relación es decreciente. La principal diferencia entre coeficientes, es la sensibilidad que tienen para detectar ciertos tipos de relaciones.

Por un lado, el coeficiente de correlación de Pearson (r) es más sensible a relaciones lineales, mientras que, Kendall (τ) y Spearman (ρ) son sensibles a relaciones no lineales. Respecto a los dos últimos coeficientes de correlación, Kendall (τ) y Spearman (ρ), este último es más utilizado en la literatura, además, Spearman (ρ) es más sensible a errores e inconsistencias en los valores medidos, y usualmente tiene un valor más alto que Kendall (τ). Información sobre el cálculo de los coeficientes de correlación se puede encontrar en el Apéndice.

Por último, si bien este método de validación es computacionalmente eficiente de realizar, además de que, intuitivamente, pareciera ser un buen método para evaluar la representación de relaciones entre palabras en los word embeddings, presenta varios problemas que resolver. La mayoría de estos problemas son descritos en [1] y [14], pero los más importantes están relacionados con la construcción de los datasets, principalmente, lo subjetivo que puede ser calificar la similitud entre palabras y la confusión que existe entre similitud semántica o relación semántica entre conceptos al momento de crear datasets (por ejemplo, "auto" y "tren" son palabras semánticamente similares, pero "auto" y "calle" son palabras relacionadas semánticamente), adicionalmente, en [37] se discute que el uso de distancia coseno no es adecuado en algunos casos.

2.3.1.2. Analogías

El uso de analogías como método de validación de word embeddings es, junto con similitud semántica, uno de los más utilizados en la literatura, y, de cierto modo, ambos métodos de evaluación son similares. Este método de validación utiliza analogías de palabras, las cuales se componen de cuatro palabras – a, b, c, d – que se relacionan de forma tal que:" 'a' es similar a 'b', de la misma forma que 'c' es similar a 'd' " denotado como $a : b :: c : d$. La Tabla 2.2 muestra algunos ejemplos de analogías, y la clase de relación a la que se asocia.

	Analogía			
Relación	a	b	c	d
Capital-País	París	Francia	Berlín	Alemania
Nombre-Nacionalidad	Franco	español	Stalin	soviético
Nombre-Profesión	Mozart	compositor	Dante	poeta
Prefijo -mente	sano	sanamente	costoso	costosamente
V. infinitivo-gerundio	abrir	abriendo	gastar	gastando

Tabla 2.2: Ejemplo de contenido de un dataset de analogías.

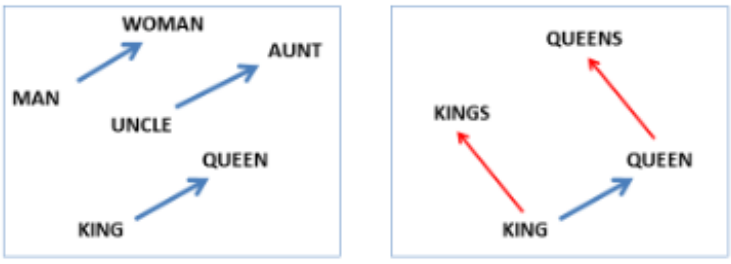

Figura 2.1: Relación entre palabras de una analogía, dentro de un word embedding.

El test de validación se realiza utilizando un conjunto de analogías, y consiste en predecir una de las palabras dentro de cada analogía, a partir de las palabras restantes de la misma analogía (usualmente, predecir "d", a partir de "a", "b" y "c"), donde la cantidad de predicciones correctas realizadas es la métrica para definir el desempeño de un embedding. El cómo se realiza la predicción es un tema de discusión en la literatura, pero, lo más común es suponer que la relación entre palabras dentro de la analogía se puede extraer de la geometría del embedding, haciendo un desplazamiento lineal de los vectores, como se aprecia en figura 2.1. Más formalmente, sea x una palabra, sea $\vec{x}$ el vector correspondiente a la palabra x dentro del embedding y sea la relación $a : b :: c : \mathrm{d}$. Se puede suponer que:

$$\vec{b} - \vec{a} + \vec{c} \approx \vec{\mathrm{d}} \tag{2.2}$$

Puesto que es difícil que la expresión anterior se cumpla de forma exacta, al momento de realizar las predicciones, se busca el vector más similar. Este método, definido en [25], se denomina 3CosAdd:

$$\vec{\mathrm{d}^*} = \underset{\vec{\mathrm{d}}}{\operatorname{argmax}} \frac{\vec{\mathrm{d}} \cdot (\vec{b} - \vec{a} + \vec{c})}{||\vec{\mathrm{d}}||||(\vec{b} - \vec{a} + \vec{c})||} \tag{2.3}$$

equivalente a tomar el vector con mayor similitud coseno al vector predicho inicialmente. Alternativamente, se puede definir otro método a partir de la ecuación 2.3, denominado 3CosMul[23]:

$$\vec{\mathrm{d}^*} = \underset{\vec{\mathrm{d}}}{\operatorname{argmax}} \frac{cos(\vec{\mathrm{d}}, \vec{c})cos(\vec{\mathrm{d}}, \vec{b})}{cos(\vec{\mathrm{d}}, \vec{a}) + \varepsilon} \quad , \quad \varepsilon = 0{,}001 \tag{2.4}$$

Respecto a la efectividad que tienen los dos métodos anteriores para predecir la palabra faltante dentro de una analogía, de los experimentos en [23] [29], se concluye que 3CosMul permite realizar una mayor cantidad de predicciones correctas, comparado con 3CosAdd.

Los métodos de predicción descritos, son los más utilizados en la literatura al momento de evaluar word embeddings a partir de analogías, sin embargo, existen métodos alternativos con los cuales realizar las evaluaciones:

- **PairDir** [23], método que también deriva de 3CosAdd, pero qué da más importancia a la dirección de los vectores que representan la relación entre los pares de palabras en la analogía, utilizando la notación de la expresión 2.2, se da más importancia a la dirección de los vectores $\vec{b} - \vec{a}$ y $\vec{\mathrm{d}} - \vec{c}$.

$$\vec{\mathrm{d}^*} = \underset{\vec{\mathrm{d}}}{\operatorname{argmax}}(cos(\vec{\mathrm{d}} - \vec{c}, \vec{b} - \vec{a})) \tag{2.5}$$

- **LRCos** [12] es un algoritmo basado en aprendizaje supervisado, el cual se entrena a partir de pares de palabras que compartan la misma relación (por ejemplo, relación capital-país), y de esta forma, aprender la relación existente entre los pares de palabras.

De los experimentos de [29], LRCos es más eficaz a la hora de realizar predicciones, comparado con 3CosMul y 3CosAdd.

- **Analogy Space Evaluation** [9], a diferencia de los anteriores, no es un método de predicción, si no que es una forma distinta de utilizar analogías como método de validación, la motivación de este método, es lo costoso que puede ser realizar predicciones a partir de 3CosAdd o 3CosMul. Se definen cuatro métricas a partir de los vectores presentes en la analogía: Cos, Euc, $N\text{-}Cos$ y $N\text{-}Euc$, siendo $N\text{-}Cos$ y $N\text{-}Euc$ iguales a Cos y Euc respectivamente, pero utilizando vectores unitarios.

$$Cos. = \frac{(\vec{b} - \vec{a})(\vec{\mathrm{d}} - \vec{c})}{||\vec{b} - \vec{a}||||\vec{\mathrm{d}} - \vec{c}||} \quad , \quad Euc. = \frac{||(\vec{b} - \vec{a}) - (\vec{\mathrm{d}} - \vec{c})||}{||\vec{b} - \vec{a}|| + ||\vec{\mathrm{d}} - \vec{c}||} \tag{2.6}$$

Finalmente, utilizar analogías como método de validación es una buena idea para evaluar la distribución de palabras dentro del embedding o evaluar alguna propiedad específica entre palabras, puesto que es un método computacionalmente eficiente y que presenta cierto grado de correlación con ciertas tareas en NLP [36] [34]. Sin embargo, hay que tener ciertas consideraciones, ya que, al igual que el test de similitud, el test de analogía también presenta subjetividad, puesto que las analogías están ligadas al razonamiento y lógica humana, además, producto de cómo son construidos los embeddings, puede ocurrir que estos encuentren relaciones distintas a las esperadas por los humanos. Por otro lado, en [29] se exponen las debilidades de realizar predicciones suponiendo que la relación entre pares de palabras se expresan como la diferencia entre los vectores correspondientes.

2.3.1.3. Outlier detection

A diferencia de los métodos mencionados anteriormente, *outlier detection* es un método relativamente nuevo, definido en [5] (aunque en [32] se define un método similar). Este método evalúa cómo se comparan las palabras entre ellas dentro de un embedding, específicamente, se evalúa la capacidad que tienen los embeddings de generar clusters semánticos, es decir, grupos de palabras con significados similares. La idea de este método es la de distinguir la palabra anómala (outlier), desde un punto de vista semántico, dentro de un conjunto de palabras que comparten alguna característica, por ejemplo, sea el conjunto de palabras {manzana, piña, frutilla, libro, plátano}, el outlier del conjunto corresponde a la palabra "libro". La Tabla 2.3 presenta otros conjuntos de palabras que pueden ser utilizados con este método.

Formalmente, el proceso de evaluación consiste en, dado un conjunto de palabras, determinar qué tan semejantes son las palabras del conjunto, omitiendo una de ellas. En este sentido, la mayor semejanza entre las palabras se logra al omitir el outlier. Para la realización de la evaluación, primero se define un valor de compacidad, el cual refleja que tan compacto es un conjunto de palabras en el word embedding. Sea $c(w)$ el valor de compacidad del conjunto $W\backslash\{w\}$ respecto al word embedding a evaluar como:

$$c(w) = \frac{1}{n(n-1)} \sum_{w_{\mathrm{i}} \in W\backslash\{w\}} \sum_{\substack{j \neq \mathrm{i} \\ w_j \in W\backslash\{w\}}} sim(w_{\mathrm{i}}, w_j) \tag{2.7}$$

Cluster principal	Internet Explorer	automoción	lactulosa	PDF
	Safari	minería	sucralosa	SVG
	Microsoft Edge	Shopping	maltosa	GIF
	Chromium	horticultura	Maltitol	CSS
	Konqueror	robótica	Isomalta	XML
	Firefox	manufactura	sacarosa	XHTML
	Opera	producción cinematográfica	celobiosa	HTML
	Google Chrome	construcción naval	trehalosa	SQL
Outliers	Emacs	anime	glicerol	XSLT
	Bloc de notas	sericicultura	goma tragacanto	MP3
	iWeb	sericicultura	gas CS	factor de forma
	Daemon Tools	tormenta	vitamina	iPad3
	Apertium	Le ballon rouge	E-105	Olimpiadas
	Gettext	Captain EO	Oxireno	UTC

Tabla 2.3: Ejemplos de conjuntos de pruebas.

dónde $sim(w_{\text{i}}, w_j)$ es una función de similitud entre los vectores del word embedding, correspondientes a las palabras w_{i} y w_j (usualmente se utiliza la función de similitud coseno). Respecto al outlier dentro de un conjunto W, este se destaca por sobre el resto de palabras al calcular el valor de compacidad. Puesto que la palabra outlier se diferencia del resto, esto se debería reflejar en el valor de compacidad, ya que, si $w \in W$ es la palabra outlier, entonces $c(w)$ debería ser el mayor posible.

A partir del valor de compacidad $c(w)$, se definen dos mediciones: *Outlier Position* (OP) y *Outlier Detection* (OD). Sea un conjunto de palabras W, compuesto por n palabras y un outlier, OP corresponde a la posición del outlier al ordenar las palabras según el valor de compacidad, el cual puede estar entre 0 y n (posición 0 indica el valor más bajo de compacidad). Mientras que OD se define como 1, si el outlier es correctamente detectado (el valor OP debería outlier es n), o 0, si el outlier no ha sido detectado. El desempeño de un embedding en este test, dado el conjunto de prueba D (compuesto por $|D|$ conjuntos de palabras), está dado por el *accuracy* (Acc) y el *Outlier Position Percentage* (OPP):

$$\text{Acc} = \frac{\sum_{W \in D} \text{OD}(W)}{|D|} \tag{2.8}$$

$$\text{OPP} = \frac{\sum_{W \in D} \frac{\text{OP}(W)}{|W|-1}}{|D|} \tag{2.9}$$

siendo Acc el porcentaje de acierto que logra el embedding en la detección de outlier, mientras que, OPP es una medida de lo cerca que estuvo el embedding de detectar correctamente los outliers.

Como fue mencionado anteriormente, este método de evaluación es relativamente nuevo, por lo que aún no se han realizado estudios exhaustivos de sus capacidades y limitaciones al evaluar word embeddings. De todas formas, un aspecto que puede ser discutido, es la medida

de similitud utilizada para deducir si una palabra es outlier (usualmente se utiliza distancia coseno). Por otro lado, en [36] se hace mención de algunos problemas con este método, como la subjetividad que presenta el test (aunque en el mismo trabajo se menciona que esto no es suficiente para influenciar los resultados).

2.3.1.4. Cross-matching

Un método relativamente reciente, propuesto en [20], y basado en el trabajo [30], es originalmente un método utilizado en el área de la medicina para comparar muestras de sangre. En el contexto de validación de word embeddings, cross-matching consiste en evaluar dos word embeddings generados a partir de un mismo corpus, y determinar cuál es la correlación que existe entre ambos, siendo, una baja correlación, indicador de que ambos embeddings utilizan características distintas del corpus. De forma más específica, durante el test de cross-match se asume la hipótesis nula: ambos embeddings son idénticos, es decir, sean W y V word embeddings generados desde el mismo corpus, la hipótesis nula H_0 establece que $W = V$.

Asumiendo la hipótesis nula, el experimento se define a continuación. Dados W y V word embeddings generados desde el mismo corpus, se define $\{w_1, ..., w_n\} \sim W$ y $\{v_1, ..., v_n\} \sim V$ conjuntos de vectores elegidos aleatoriamente, y sea el estadístico de cross-match C, una función del conjunto $D = \{w_1, ..., w_n, v_1, ..., v_n\}$. Entonces, el test de cross-match consiste en encontrar la permutación σ que minimiza la ecuación:

$$\text{Match}(\sigma) = \sum_{\text{i}=1}^{N} \text{d}(D_{\text{i}}, D_{\sigma(\text{i})}), N = |D| = 2n \tag{2.10}$$

donde $\forall \text{i}, \text{i} \neq \sigma(\text{i})$, d es una medida de distancia entre vectores y el estadístico C queda definido como la cantidad de pares $(\text{i}, \sigma(\text{i}))$ tal que los vectores D_{i} y $D_{sigma(\text{i})}$ pertenecen a distintos word embeddings. Es importante destacar que, resolver este problema de optimización es equivalente a resolver el problema de matching no bipartito.

La motivación detrás de calcular el estadístico C proviene del supuesto de la hipótesis nula, puesto que se asume correcta en un comienzo, la permutación obtenida a través de minimizar la ecuación anterior, debería ser similar a una elección aleatoria. Considerando lo anterior, se puede concluir que el valor de C también debería ser un valor aleatorio. Para determinar si la hipótesis nula se cumple, se procede a calcular el *p-value* del estadístico C, lo cual corresponde a la probabilidad de que C sea menor o igual al valor alcanzado en el experimento. Si el p-value es bajo, esto implica que el valor obtenido para C es poco probable que sea resultado de una elección aleatoria, con lo cual, se puede negar la hipótesis nula, entonces $W \neq V$. En el contexto de word embeddings, un p-value bajo, implica que los embeddings comparados, utilizan distintas características del corpus con el cual se crearon.

El trabajo original, en el que se propone este método de validación, concluye que la principal utilidad de este método es el análisis de word embeddings a través de distintos lenguajes. Por otra parte, en el mismo trabajo se establece que la principal debilidad de este método, es la gran demanda de recursos que se necesitan para resolver el problema de optimización. Adicionalmente, para que los resultados obtenidos a través de este test sean

significativos, es necesario realizarlo más de una vez. Por el momento, no se han encontrado otros trabajos que estudien la utilidad y debilidades de este método de validación.

2.3.2. Validación extrínseca

A diferencia de los métodos de validación intrínsecos, los métodos de validación extrínsecos evalúan la capacidad que tiene un embedding para resolver tareas en NLP, al ser utilizados en algoritmos de machine learning. Si bien, es usual utilizar este tipo de métodos al evaluar las capacidades de un word embedding, estos métodos no deberían ser tratados como una forma absoluta de evaluar embeddings. Esto se debe a que, al parecer, no existe una correlación entre los resultados obtenidos a través de distintos métodos extrínsecos. En relación a lo anterior, en [32] se hace mención de cómo ciertos métodos extrínsecos, parecieran favorecer distintos tipos de embeddings.

A continuación se presentan algunos ejemplos de tareas que pueden ser utilizadas como métodos extrínsecos de validación:

- **Clasificación de texto.** Como el nombre lo indica, esta tarea consiste en la clasificación de distintos textos. Esta tarea puede variar dependiendo de qué es lo que se quiera clasificar, ejemplos de esto son las tareas de Análisis de Sentimiento (detectar si un texto presenta un sentimiento positivo o negativo frente a algo) y Detección de Metáforas.

- **Named Entity Recognition.** La idea de esta tarea es identificar entidades con nombre dentro de textos, como por ejemplo, nombres de personas, instituciones, marcas comerciales, etc.

- **Part-of-speech tagging.** Esta tarea trata de reconocer y etiquetar las palabras presentes en un texto según su categoría gramatical, las que pueden ser pronombres, verbos, sustantivos, adjetivos, etc.

Finalmente, cabe destacar que, a partir de la definición de los métodos de validación extrínsecos, cualquier tarea perteneciente al procesamiento de lenguaje natural puede ser utilizada como método de validación.

Capítulo 3

Herramienta de Evaluación

En este capítulo se detalla la implementación y desarrollo de una herramienta de evaluación de word embeddings, uno de los objetivos principales de este trabajo. Este capítulo presenta una descripción general de la herramienta desarrollada, además de una descripción de los dataset y métricas presentes en los distintos métodos de evaluación implementados.

3.1. Descripción general

La implementación de esta herramienta de evaluación fue realizada utilizando el lenguaje de programación Python, en conjunto con el IDE PyCharm. El principal motivo por el que se decidió utilizar Python, fue por su facilidad de uso y la existencia de librerías para el manejo de vectores, y librerías orientadas a NLP y redes neuronales. Las principales librerías que fueron utilizadas son Numpy, scikit-learn, Gensim y PyTorch.

El resultado final es un programa, compuesto de cinco módulos, cada módulo corresponde a la implementación de un método de validación, cuatro métodos intrínsecos y uno extrínseco, además de los datasets pertinentes. Este programa permite la evaluación de word embeddings, utilizando alguno de los métodos implementados. Como un producto adicional del desarrollo de esta herramienta, también se creó un nuevo dataset para la evaluación de word embeddings por analogías de palabras.

Un aspecto importante de esta herramienta, es el manejo de los datasets utilizados en los procesos de evaluación. Para simplificar el proceso de evaluación, la herramienta desarrollada cambia las letras mayúsculas por minúsculas durante las evaluaciones. Esto es debido a que existen word embeddings que no utilizan letras mayúsculas dentro de su vocabulario (por ejemplo, 'Alemania' no existe en el vocabulario, pero si existe 'alemania'). Además, durante la evaluación de un embedding, se omiten aquellas palabras que no aparezcan dentro del vocabulario del embedding. Lo anterior facilita la adición de nuevos dataset de evaluación, fuera de los presentados en este trabajo. Finalmente, se incluye la opción para realizar una evaluación comparativa de word embeddings. En este formato de evaluación, se omiten las palabras del datasets que no aparezcan en los vocabularios de todos los embeddings, de esta forma, los resultados de la evaluación entre embeddings pueden ser mejor comparados.

La herramienta de evaluación desarrollada, así como la forma de uso y los datasets empleados durante el proceso de evaluación, pueden encontrarse en github[1].

3.2. Evaluación Intrínseca

A continuación se presenta el desarrollo e implementación de los distintos métodos de validación intrínseca descritos en la sección anterior: Similitud semántica, Analogías de palabras, Outlier detection y Cross-match. Principalmente se describen los datasets y métricas que utiliza la herramienta de evaluación al aplicar los métodos mencionados.

3.2.1. Similitud semántica

Como fue mencionado en el Capítulo 2, existen trabajos donde se han evaluado word embeddings a través de similitud semántica, por lo que se contaba con un dataset (SemEval-2017) con el cual implementar este método de validación. Aún así, buscamos datasets alternativos, con los cuales entregar una mayor variedad de resultados al comparar el desempeño de distintos embeddings. Respecto a los datasets encontrados, todos corresponden a la traducción al español de datasets utilizados en la validación de word embeddings para lenguaje inglés.

A continuación se presenta una descripción de los dataset encontrados:

- **Multi-SimLex** [35] dataset compuesto por 1888 pares de conceptos (pares formados por más de dos palabras), los cuales pueden tener un valor de similitud semántica entre 0 y 6. El trabajo original cuenta con una traducción al español, creado para ser utilizado en la validación de word embeddings multilinguals o cross-lingual.

- **SimLex-999** [22] dataset compuesto por 999 pares de palabras, los cuales pueden tener un valor de similitud semántica entre 0 y 10. El dataset original solo se desarrollo en inglés, pero [2] y [13] realizan una traducción al español.

- **SemEval-2017** [6] dataset compuesto por 500 pares de conceptos (los cuales pueden consistir de más de dos palabras), los cuales pueden tener un valor de similitud semántica entre 0 y 4. El trabajo original cuenta con una traducción al español, como parte del task 2 en SemEval-2017 (Multilingual and Cross-lingual Semantic Word Similarity).

- **WordSim-353** [16] dataset compuesto por 353 pares de palabras, los cuales pueden tener un valor de relación semántica entre 0 y 10. El dataset original solo se desarrollo en inglés, pero [2] y [21] realizan una traducción al español.

- **RG-65** [31] (acrónimo de Rubenstein y Goodenough) dataset compuesto de 65 pares de palabras, los cuales pueden tener un valor de similitud semántica entre 0 y 4. El dataset original solo se desarrolló en inglés, pero [7] y [2] realizan una traducción al español.

[1] `https://github.com/Rukua95/Spanish_Word_Embedding_Evaluations`

- **MC-30** [26] (acrónimo de Miller y Charles) dataset se compone de 30 pares de palabras, los cuales pueden tener un valor de similitud semántica entre 0 y 4. El dataset original solo se desarrollo en inglés, pero [2] y [21] realizan una traducción al español.

De los datasets expuestos, los datasets elegidos para ser utilizados durante el proceso de evaluación son: Multi-SimLex, SemEval-2017, WordSim-353 traducido en [21], RG-65 traducido en [7] y MC-30 traducido en [21]. El motivo por el cual no se utilizaron ciertas traducciones, es debido a que no se logró obtenerlas [13], o debido a que el proceso de traducción no está del todo claro [2] (en este caso, no hay información completa sobre el proceso de traducción de los pares de palabras y cómo se refleja en el valor de similitud).

Finalmente, respecto a las métricas utilizadas, se utilizó la función de distancia coseno para medir la similitud entre vectores. Por otro lado, para los resultados, se utilizaron las tres métricas descritas en Capítulo 2: Pearson (r), Spearman (ρ) y Kendall (τ). El uso de todas estas métricas es para obtener una mejor comparación de resultados entre distintos word embeddings, además de que calcular estos resultados no es muy demandante en recursos (aunque esto puede no ocurrir si se utilizan datasets más grandes).

3.2.2. Analogías de palabras

Al igual que para el método anterior, también existen trabajos donde word embeddings para el español fueron evaluados a través del uso de analogías de palabras, por lo que desde un principio se contó con un dataset para implementar este método de evaluación. De todas formas, realizamos una búsqueda de dataset alternativos con los cuales tener una mayor variedad de resultados al momento de comparar word embeddings, sin embargo, no se logró encontrar otros dataset en español con los cuales realizar la evaluación.

El dataset con el cual se contaba inicialmente, corresponde a la traducción al español [8] del dataset presentado en [24], denominado también Google Analogy, el cual cuenta con 8869 analogías semánticas, repartidas en 5 clases de analogías, y 10675 analogías sintácticas, repartidas en 9 clases de analogías.

Tipo de analogía		Clase	Cantidad	Ejemplo
Semántica	GA1	Capitales Conocidas	420	Atenas : Grecia :: Bagdad : Irak
	GA2	Capitales del mundo	4369	Abuja : Nigeria :: Accra : Ghana
	GA3	Moneda	752	Argelia : dinar :: Angola : kwanza
	GA4	Ciudad y estado (EEUU)	2182	Chicago : Illinois :: Houston : Texas
	GA5	Hombre-Mujer	380	chico : chica :: hermano : hermana
Sintáctica	GA6	Adjetivo-Adverbio	552	usual : usualmente :: raro : raramente
	GA7	Opuestos	380	posible : imposible :: lógico : ilógico
	GA8	Presente participio	930	baile : bailando :: volar : volando
	GA9	Nacionalidad	1521	Ucrania : Ucranio :: India : Indio
	GA10	Pasado	1190	leyendo : leer :: tomando : tomó
	GA11	Plural	1332	plátano : plátanos :: nube : nubes
	GA12	Verbos plurales	756	escribir : escribe :: caminar : camina

Tabla 3.1: Descripción y ejemplos de traducción de dataset Google Analogy.

La tabla 3.1 entrega una pequeña descripción de cómo está estructurado la traducción del dataset Google Analogy, siendo la principal diferencia respecto al dataset original, la ausencia de las clases de analogías "Superlativos" y "Comparativos", debido a que estas palabras no tienen una traducción directa al español.

Si bien, no encontramos datasets distintos a Google Analogy, basado en el desarrollo de BATS (Bigger Analogy Test Set [19]) creamos un dataset propio, esto fue motivado por la reducida cantidad de clases presentes en Google Analogy y el desbalance que hay entre clases de un mismo tipo de analogía (capitales-países es cerca del 50 % de las analogías semánticas). El dataset creado, el cual nombramos SATS (Spanish Analogy Test Set), detallado en tabla 3.2, se compone de 29 clases de analogías, las que a su vez, se distribuyen en 4 tipos: derivaciones morfológicas, inflexiones morfológicas, lexicográficas y enciclopédicas.

Tipo analogía		Clase		Ejemplo
Sintáctica	Derivación	D1	Prefijo anti-	balas:antibalas
		D2	Prefijo des-	afina:desafinar
		D3	Prefijo in-	acción:inacción
		D4	Sufijo -able	aceptar:aceptable
		D5	Sufijo -cion	evaluar:evaluación
		D6	Sufijo -isimo	corto:cortísimo
		D7	Sufijo -ito	cielo:cielito
		D8	Sufijo -mente	común:comúnmente
		D9	Sufijo -miento	padecer:padecimiento
		D10	Pais y gentilicio	Bélgica:belga
	Inflexión	I1	Gerundio y participio	creando:creado
		I2	Infinitivo y gerundio	bailar:bailando
		I3	Infinitivo y participio	caer:caído
		I4	Pretérito perf. y futuro simple singular	conduje:conduciré
		I5	Presente y futuro simple singular	corto:cortaré
		I6	Presente y pretérito perf. singular	cierro:cerré
		I7	Singular y plural presente	corro:corremos
		I8	1^a y 3^a pers. presente	exploro:explora
		I9	Plural -s	amigo:amigos
		I10	Plural -es	dios:dioses
Semántica	Enciclopédica	E1	Capital y pais	Austria:Viena
		E2	Pais e idioma	Belice:inglés
		E3	Nombre y pais	Cesar:romano
		E4	Nombre y ocupacion	Dante:poeta
		E5	Masculino y femenino	toro:vaca
		E6	Ciudad y provincia (Chile)	Illapel:Coquimbo
		E7	Ciudad y estado (EEUU)	Joliet:Illinois
	Lexicográfica	L1	Sinónimos	cálido:suave/moderado/templado
		L2	Antónimo	peligroso:seguro

Tabla 3.2: Descripción y ejemplos de dataset SATS.

Cabe destacar que el dataset creado, SATS, presenta ciertas diferencias respecto a Google Analogy. La principal diferencia, es la posibilidad de que una analogía puede estar compuesta por más de 4 palabras. Por ejemplo, sea "Canadá:inglés/francés :: Kazajistán:kazajo/ruso" una de las analogías presente en SATS, durante el proceso de predicción, las palabras "kazajo" y "ruso" son igualmente válidas. Respecto a la estructura del nuevo dataset, todas las clases están formadas con 50 pares de palabras, con lo cual, se cuentan con 2450 analogías distintas

por clases. Pero, debido a las dificultades para construir SATS, no fue posible crear un dataset que fuera balanceado a través de los distintos tipos de analogías.

La creación de este nuevo dataset se realizó de distintas formas, las cuales dependen principalmente del tipo de clase de analogía que se construyó:

- Para el caso de analogías de tipo derivación morfológica, tanto para el caso de sufijos y prefijos, se inició por una selección de prefijos y sufijos presentes en WikCionario[2]. Una vez seleccionados los prefijos y sufijo, se extrajeron las palabras correspondientes desde la misma página web (WikCionario presenta las palabras con y sin sufijo, e igualmente para los prefijos).

- Las analogías de tipo inflexión morfológica se separan en dos grupos: tiempos verbales y plurales. Para crear las analogías de tiempos verbales, se eligió un conjunto de verbos en infinitivo, para luego buscar los mismos verbos en distintos tiempos verbales desde la página web de la RAE[3], esto resulto en la creación de más de 400 clase de analogías, en cambio, las analogías de plurales fueron construidas manualmente, utilizando la página de la RAE como referencia.

- Para las analogías enciclopédicas, la mayoría se construyó utilizando la información presente en la página web Wikipedia. Un caso particular, fue la clase de analogías masculino-femenino, en la cual se eligió un conjunto de palabras de género masculino, para luego determinar las palabras correspondientes de género femenino.

- Para las dos analogías lexicográficas del dataset, se eligió un conjunto de palabras, para luego extraer sinónimos y antónimos desde un diccionario presente en internet[4].

Complementando lo anterior, los conjuntos de palabras con los que se construyó el dataset corresponden a la traducción de las palabras presentes en las clases equivalentes desde BATS. Para la obtención de información desde las distintas páginas web mencionadas, creamos distintos programas, los cuales extrajeron la información desde internet, a su vez, también creamos un programa que verificó que las palabras dentro del nuevo dataset estuvieran presentes en la RAE.

Para la medición del desempeño de los word embeddings, se implementaron tres métricas: 3CosMul, 3CosAdd y Analogy Space Evaluation. Las primeras dos fueron elegidas por ser las más utilizadas en la literatura, mientras que la tercera, se eligió debido a lo poco demandante que es el proceso de cálculo. Al utilizar 3CosMul y 3CosAdd, la herramienta de evaluación entrega la precisión al predecir correctamente la palabra faltante de la analogía (top-1), y predecir dentro de las mejores cinco opciones (top 5). Para Analogy Space Evaluation se calculó el promedio de Cos y Euc, a través de las analogías de una misma clase.

[2]https://es.wiktionary.org/wiki/Wikcionario:Portada
[3]Real Academia Española https://www.rae.es/
[4]https://www.wordreference.com/sinonimos/

3.2.3. Outlier detection

A diferencia de los dos métodos descritos anteriormente, el uso de outlier detection como método de validación es más reciente, por lo que no hay una gran variedad de datasets. De lo investigado, solo se encontraron dos dataset, los cuales seguían la misma estructura:

- **8-8-8 Dataset** [5] un dataset pequeño, compuesto de 8 conjuntos de prueba, donde cada uno de los conjuntos contienen 8 conceptos en el grupo principal y 8 conceptos en el grupo outlier.

- **WordSim-500** [3] mucho más grande que el dataset anterior, se divide en cinco lenguajes distintos (no necesariamente con el mismo contenido), entre ellos el español. El dataset en español se compone de 500 conjuntos de prueba, donde cada uno de los conjuntos contiene 7 u 8 conceptos en el grupo principal, y hasta 6 conceptos en el grupo outlier.

Para el proceso de evaluación a partir de este método se utiliza la sección en español del dataset WordSim-500. Un aspecto importante a destacar para este método de evaluación, es el caso en que algunas palabras sean omitidas al evaluar word embeddings. Como se mencionó inicialmente, las palabras que no se encuentren en el vocabulario del word embedding a evaluar son omitidas, lo que puede reducir el tamaño del grupo principal u outlier de los conjuntos de prueba del dataset. Para estos casos, se utiliza el mismo procedimiento que en [3]. Un conjunto de prueba es utilizado, si la cantidad de palabras del grupo outlier es mayor a cero y si la cantidad de palabras del grupo principal es mayor a uno. Finalmente, la métrica utilizada como medida de similitud entre vectores, fue la función de similitud coseno.

3.2.4. Cross-match

A diferencia de los métodos descritos anteriormente, cross-match no requiere de un dataset con el cual realizar evaluaciones, por lo que el tiempo de desarrollo de este método fue mucho más reducido. Al igual que en métodos anteriores, se utilizó la función de similitud coseno, como medida de distancia entre vectores. La implementación de este método sigue el procedimiento explicado en el trabajo original [20].

1. Se extrae un conjunto inicial de vectores desde cada uno de los word embeddings a evaluar.

2. De cada uno de los conjuntos generados, se extrae un subconjunto de vectores de forma aleatoria. Este par de conjuntos se unen en un solo conjunto.

3. Para cada par de vectores dentro del nuevo conjunto, se calcula la distancia coseno.

4. A partir de los valores de distancia calculados en el paso anterior, se determina la permutación que minimiza la ecuación 2.10, y se calculó el estadístico C.

5. Finalmente, a partir del estadístico C, se calcula el p-value para el test.

Para mejorar la precisión de los resultados, el paso 2 hasta el paso 5 son repetidas varias veces. El resultado final de la evaluación es el promedio de los p-value obtenidos en cada

iteración. Para simplificar el uso de este método en la herramienta de evaluación, se mantuvo fijo la cantidad de vectores que se extraen y la cantidad de iteraciones realizadas. Inicialmente se extraen 10.000 vectores de cada word embeddings, se realizan 100 iteraciones, y en cada iteración se seleccionan 200 vectores aleatorios de cada conjunto inicial.

Para la resolución del problema de optimización asociado a este método, se utilizó la librería NetworkX de Python, el cual entrega métodos para resolver el problema de matching bipartito (el problema de optimización presentado es equivalente al problema de matching bipartito).

3.3. Evaluación Extrínseca

A continuación se presenta el desarrollo de la validación extrínseca implementada para la herramienta de evaluación, la cual consiste en la resolución de tareas de clasificación de textos, específicamente, las tareas de clasificación A y B definidas en [15]. Para ello implementamos dos métodos: clasificación utilizando el promedio de vectores representantes correspondientes a las palabras presentes en los textos, y clasificación utilizando redes neuronales. Principalmente se expone el contexto de las tareas de clasificación a utilizar, así como el dataset asociado.

3.3.1. Contexto

Durante el año 2016, el gobierno de Chile inició un proceso participativo, para delinear qué cosas debería tener una nueva constitución. La primera fase de este proceso consistió en pequeñas asambleas, en donde los participantes tuvieron que acordar cuáles eran los conceptos constitucionales más importantes, y escribir un argumento que justificara la importancia de dicho concepto. Al finalizar esta fase, se obtuvo un dataset de más de 200.000 argumentos políticos.

Cabe destacar que, los conceptos que los participantes podían escoger y argumentar, se dividían en cuatro tópicos: Valores (**Values**, 37 conceptos), Derechos (**Rights**, 44 conceptos), Deberes (**Duties**, 12 conceptos) e Instituciones (**Institutions**, 21 conceptos), adicionalmente, los participantes contaban con la opción de incluir nuevos conceptos, fuera de los predefinidos inicialmente. Estos nuevos conceptos, definidos por los participantes de la asamblea, se les denominó conceptos abiertos. Una vez finalizada la primera fase, gran parte de los conceptos abiertos fueron categorizados en los conceptos predefinidos por el gobierno. En caso de que los conceptos no pudieran ser categorizados, estos se clasificaban dentro de nuevas clases de conceptos, o se declaraban como inclasificables.

Conceptos	Argumentos
Justicia	la justicia no solo relacionada con el derecho penal sino relacionada con la justicia social para todos y todas mayores niveles de igualdad
Igualdad	es prioritario asegurar igualdad sustantiva de derechos que vaya más allá de las oportunidades y condiciones de acceso a hombres y mujeres independiente de su género grupos lgtb país de origen diversidad étnica ante la ley etcétera
Estado laico	es necesario reafirmar una definición laica esta definición no se observa en la práctica ya que a la hora de legislar aún aparece en los argumentos visiones desde conceptos religiosos principalmente de la iglesia católica de los legisladores como en las leyes sobre equidad de género aborto

Tabla 3.3: Ejemplos de conceptos del tópico Valores y sus argumentos.

3.3.2. Tareas de clasificación

Previo a definir las tareas de clasificación, es necesario establecer la notación a utilizar: sea un tópico $T \in \{V, R, D, I\}$, se define D_G^T conjunto de pares (c, a), tal que a es un argumento para el concepto c predefinido por el gobierno dentro del tópico T, y sea D_O^T conjunto de pares (c, a), tal que a es un argumento para el concepto abierto c dentro del tópico T.

A continuación se definen las tareas de clasificación que se utilizarán como método de validación extrínseco:

- Tarea A: Se define C_G^T como el conjunto de conceptos predefinidos por el gobierno para el tópico T, y sea A_G^T el conjunto de argumentos para conceptos en C_G^T. Dado un argumento $a* \in A_G^T$, se tiene que predecir el concepto $c \in C_G^T$ tal que $(c, a*) \in D_G^T$.
- Tarea B: Dado un par $(c*, a*) \in D_O^T$ determinar el concepto $c \in C_G^T$ al cual más probablemente hace referencia el par $(c*, a*)$.

Concretamente, las tareas de clasificación a realizar consisten en: clasificar argumentos según el concepto de gobierno al que van dirigidos (tarea A), y clasificar conceptos abiertos según el concepto de gobierno más adecuado que los represente (tarea B). Cabe destacar que, puesto que los tópicos son independientes entre ellos, se tiene un total de ocho tareas de clasificación distintas, dos por cada tópico.

3.3.3. Dataset

El dataset generado a partir del proceso mencionado anteriormente, cuenta con 205,357 argumentos, de los cuales, el 10.7 % son argumentos para conceptos abiertos. De los conceptos abiertos, solo 10,263 pudieron ser catalogados en los conceptos definidos por el gobierno. Por otro lado, del resto de conceptos abiertos, solo 8,751 fueron clasificados en nuevos conceptos, mientras que el resto quedaron como inclasificables. La Tabla 3.4 presenta la distribución de argumentos, conceptos de gobierno y conceptos abiertos, a través de los distintos tópicos.

Tópico	Argumentos	Conceptos de gobierno	Conceptos abiertos
Valores	53,780	37	1,876
Derechos	53,060	44	3,712
Deberes	48,758	12	2,860
Instituciones	49,759	21	3,120

Tabla 3.4: Distribución de argumentos, conceptos de gobierno y conceptos abiertos dentro del dataset.

Un caso particular de uso del dataset, fue para la resolución de la tarea A utilizando una red neuronal LSTM. Debido a como es el funcionamiento de una red neuronal, se crearon tres datasets específicos para este método de clasificación. Estos tres dataset son una partición del dataset original, correspondientes a un dataset de entrenamiento, validación y testing. Para la separación del dataset, se consideró una partición de 80 % como entrenamiento, 10 % validación y el resto como testing. Durante la división del dataset, se mantuvo en consideración que los argumentos pertenecientes a algún concepto se distribuyeran en la proporción 8:1:1. La tabla 3.5 muestra el tamaño final de cada sección del dataset con respecto a los distintos tópicos.

Dataset	Valores	Derechos	Deberes	Instituciones	Tamaño total
Entrenamiento	37.508	36.346	33.173	32.780	139.807
Validacion	4.690	4541	4.145	4097	17.473
Testing	4.705	4.565	4.137	4.107	17.514

Tabla 3.5: Tamaño de las distintas sección utilizadas en la clasificación a partir de redes neuronales.

3.3.4. Métodos de clasificación

Para realizar la clasificación de textos, se utilizaron dos métodos de clasificación a partir de word embeddings: a través del uso de vectores promedio, y utilizando redes neuronales. La clasificación a partir de vectores promedio, consiste en promediar los vectores que representan a las palabras presentes en un texto, y utilizar este vector promedio como el vector representante del texto. Del mismo modo, se calcula un vector representante de las clases, a partir de promediar los vectores representantes de las palabras que componen el nombre real de la clase. Luego, un texto se clasificará dentro de una clase, si el vector representante de la clase, es el más similar al vector representante del texto. Para este caso, se utilizó la distancia coseno como medida de similitud entre vectores.

El segundo método de clasificación, implica el uso de redes neuronales, específicamente, redes neuronales recurrentes, las cuales están diseñadas para trabajar con datos secuenciales, como por ejemplo, textos. Para este trabajo, se utilizó el mismo tipo de red neuronal para todas las tareas de clasificación, una red LSTM (Long Short-Term Memory), la cual puede ser utilizada directamente con la librería PyTorch para Python.

Capítulo 4

Evaluación de Word Embeddings

Como se mencionó al inicio de este trabajo, uno de los objetivos principales es evaluar distintos word embeddings para el lenguaje español, generados a partir de distintos corpus de texto y distintos algoritmos. Estos embeddings fueron evaluados utilizando la herramienta descrita en el capítulo anterior. En este capítulo se exponen los resultados obtenidos durante el proceso de evaluación y su correspondiente análisis. También se exponen detalles del proceso de evaluación.

4.1. Word Embeddings

Previo a exponer los resultados del proceso de evaluación, es necesario describir los word embeddings con los que se trabajó. En total se evaluaron 8 word embeddings pre-entrenados, los cuales pueden ser descargados desde github[1] [2] [3], a su vez, también se puede encontrar información sobre el proceso de entrenamiento.

Estos word embeddings se diferencian principalmente en el corpus y algoritmos utilizados durante su creación. Los algoritmos utilizados en la generación de estos word embeddings corresponden a FastText, GloVe, Word2Vec y BERT. Mientras que los corpus de texto corresponden a SUC[4], SBWC[5] y Wikipedia en español[6]. La Tabla 4.1 entrega una descripción general de los embeddings evaluados durante este trabajo. Los embeddings pueden ser descargados desde github.

Cabe destacar que algunos de los embeddings presentados en este trabajo han sido evaluados con algunos de los métodos intrínsecos descritos en Capítulo 2, específicamente, los embeddings FT-SUC(M), FT-SUC(L) y FT-SUC(NL), fueron evaluados con los métodos de similitud semántica, mientras que, FT-SBWC, fue evaluado utilizando analogías.

[1] `https://github.com/dccuchile/spanish-word-embeddings`
[2] `https://github.com/BotCenter/spanishWordEmbeddings`
[3] `https://github.com/dccuchile/beto`
[4] Spanish Unannotated Corpora `https://github.com/josecannete/spanish-corpora`
[5] Spanish Billion Word Corpus `http://crscardellino.github.io/SBWCE/`
[6] `https://archive.org/details/eswiki-20150105`

Nombre	Algoritmo	Corpus	Numero de vectores	Tamaño de vectores
FT-SUC(M)	FastText	SUC	1,313,423	100
FT-SUC(L)	FastText	SUC	1,313,423	300
FT-SUC(NL)	FastText	SUC	1,451,827	300
FT-Wiki	FastText	Wikipedia	985,667	300
GV-SBWC	Glove	SBWC	855,380	300
FT-SBWC	FastText	SBWC	855,380	300
W2V-SBWC	Word2Vec (Skip-Gram)	SBWC	1,000,653	300
BETO	BERT	SUC	—	768

Tabla 4.1: Descripción de word embeddings a evaluar.

En general, los word embeddings evaluados se diferencian a partir del algoritmo o corpus de texto utilizado para generar dichos word embeddings. Sin embargo, los word embeddings FT-SUC(M), FT-SUC(L) y FT-SUC(NL) solo se diferencian por el tamaño del vocabulario que contienen y las dimensiones de los vectores.

El caso del word embedding BETO es especial, puesto que trabaja de forma distinta al resto de embeddings. A diferencia del resto de modelos, BETO puede ser utilizado para determinar la representación vectorial de oraciones completas, a su vez, también es posible calcular representaciones vectoriales de palabras, bajo el contexto de una oración. Gracias a este último, BETO tiene la capacidad de representar palabras polisémicas utilizando más de un vector. Otro aspecto diferente de este modelo, es su vocabulario, el cual está compuesto por palabras, sub-palabras y caracteres individuales, los cuales pueden combinarse para poder representar palabras que estén fuera del vocabulario original.

En lo que respecta a este trabajo, se estudiará la capacidad que tiene BETO para representar palabras individuales, sin ningún tipo de contexto. Para el caso de palabras fuera del vocabulario de este modelo, estas se representarán como el promedio de los vectores representantes de las sub-palabras y caracteres correspondientes.

4.2. Evaluación Intrínseca

4.2.1. Resultados para Similitud semántica

Para la evaluación a partir de similitud semántica sólo se utilizó el vocabulario que compartían todos los word embeddings a evaluar. Adicionalmente, todas las letras mayúsculas fueron reemplazadas por minúsculas. Esto implica que algunos de los pares de palabras presentes en los dataset de evaluación tuvieron que ser omitidas. La Tabla 4.2 muestra el tamaño final de los datasets utilizados, además de la cantidad de pares de conceptos omitidos y repetidos.

Dataset	Pares oov[7]	Pares repetidos	Tamaño final
MultiSimLex	105	11	1772
SemEval-2017	135	0	365
WordSim-353	11	1	341
RG-65	0	0	65
MC-30	1	0	29

Tabla 4.2: Descripción de dataset usados en similitud semántica.

Respecto a los resultados de las evaluaciones, aquellos obtenidos a traves de los datasets MultiSimLex, SemEval-2017 y WS-353, utilizando las métricas establecidas en el Capítulo 3, pueden ser observados en la Tabla 4.3. Puesto que los dataset RG-65 y MC-30 son de tamaño muy reducido, se consideró que los resultados obtenidos tienen poca importancia para este trabajo. De todas formas, los resultados obtenidos a través de estos dos últimos dataset se pueden ver en el Apéndice.

	MSL			SE-17			WS-353		
Embeddings	r	ρ	τ	r	ρ	τ	r	ρ	τ
FT-SUC(M)	0.40	0.43	0.30	0.72	0.72	0.53	0.57	0.59	0.42
FT-SUC(L)	**0.46**	**0.48**	**0.34**	**0.73**	**0.75**	**0.56**	**0.58**	**0.62**	**0.44**
FT-SUC(NL)	0.44	0.45	0.32	0.71	0.73	0.54	0.57	0.60	0.42
FT-SBWC	0.43	0.44	0.31	0.69	0.70	0.52	0.56	0.59	0.42
FT-Wiki	0.43	0.44	0.31	0.71	0.73	0.55	0.57	0.60	0.43
GV-SBWC	0.42	0.44	0.31	0.66	0.68	0.49	0.33	0.32	0.22
W2V-SBWC	0.45	0.47	0.33	0.72	0.72	0.54	0.56	0.57	0.40
BETO	0.32	0.30	0.20	0.45	0.46	0.31	0.36	0.34	0.23

Tabla 4.3: Resultados para test de similitud semántica.

De los resultados de la evaluación por similitud semántica presentes en la Tabla 4.3, podemos observar que, a excepción del word embedding BETO, todos los modelos alcanzan resultados muy similares. Esto indicaría que los modelos representan casi al mismo nivel la relación de similitud entre palabras.

A partir de los resultados, podemos identificar tres word embeddings que se destacan de algún modo. En primera instancia, el word embedding FT-SUC(L) obtiene los mejores resultados para todas las métricas, a través de todos los dataset. Un segundo embedding interesante, es GV-SBWC, word embedding generado a partir del algoritmo GloVe y el corpus de texto SBWC. Este embedding obtiene resultados bajos en el dataset WS-353, respecto al resto de embeddings. Esto indica que GV-SBWC tiene una menor capacidad para representar la relación semántica entre palabras, comparado con el resto de embeddings. Por último, BETO se destaca por tener resultados mucho menores que el resto de embeddings, a través de todos los dataset.

[7]Fuera de vocabulario (out of vocabulary)

4.2.2. Resultados para Analogías de Palabras

Para el proceso de evaluación por analogía de palabras, se trabajó con los dataset de la misma manera que para similitud semántica. Todas las letras mayúsculas se cambiaron a minúsculas y se omitieron las palabras que no estuvieran en el vocabulario de todos los embeddings. Lo anterior también implicó que parte de las analogías tuvieran que ser omitidas.

La Tabla 4.4 y 4.5 muestran la cantidad de analogías que se utilizaron durante el proceso de evaluación, separadas según la clase de analogías.

	GA1	GA2	GA3	GA4	GA5	GA6
Cant. Analogías	306	524	646	209	380	552
	GA7	GA8	GA9	GA10	GA11	GA12
Cant. Analogías	380	930	1370	1190	1332	756

Tabla 4.4: Cantidad final de analogías en dataset Google Analogy.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Cant. Analogías	2450	2450	1056	2450	2450	2450	2450	2450	2450	1260
	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
Cant. Analogías	2450	2450	2450	1506	1260	1560	2256	2352	2450	2450
	E1	E2	E3	E4	E5	E6	E7	L1	L2	
Cant. Analogías	870	1560	756	600	2450	42	90	2450	2450	

Tabla 4.5: Cantidad final de analogías en dataset SATS.

Para la evaluación de los word embeddings, se utilizaron 3CosMul y Analogy Space Evaluation. Se eligieron estas métricas debido a que evalúan distintos aspectos de los word embeddings, en relación a las analogías utilizadas. El único word embedding que evaluamos de forma distinta, es BETO, para el cual solo se utilizó Analogy Space Evaluation como medida de desempeño. Esta decisión es debido a que BETO no tiene un vocabulario fijo, por lo cual, realizar predicciones utilizando 3CosMul, o cualquier otra métrica similar, podría no entregar resultados correctos.

A continuación se presentan los resultados de la evaluación por analogías de palabras. Debido a la gran cantidad de clases de analogías (12 en Google Analogy y 29 en SATS) solo se presentan los resultados promedio para cada grupo de analogías. Los resultados individuales para cada tipo de analogía se pueden encontrar en el Apéndice. La Tabla 4.6 presenta los resultados obtenidos a partir de las secciones semánticas y sintácticas del dataset Google Analogy. Las Tablas 4.8 y 4.7 contienen los resultados para las secciones semánticas y sintácticas del dataset SATS. Los resultados obtenidos a partir de las clases de analogías E6 (ciudad-provincia Chile) y E7 (ciudad-estado EEUU) no fueron considerados, debido a la poca cantidad de analogías presentes.

	Semántica				Sintáctica			
Embeddings	top1	top5	Cos	Euc	top1	top5	Cos	Euc
FT-SUC(M)	0.49	0.60	**0.45**	**0.47**	0.41	0.56	**0.37**	**0.45**
FT-SUC(L)	**0.57**	**0.67**	0.35	0.43	0.49	0.65	0.29	0.41
FT-SUC(NL)	0.56	**0.67**	0.35	0.43	**0.51**	0.65	0.30	0.41
FT-SBWC	0.55	**0.67**	0.32	0.41	0.50	**0.66**	0.27	0.40
FT-Wiki	0.51	0.61	0.30	0.41	0.48	0.62	0.26	0.39
GV-SBWC	0.54	0.64	0.33	0.42	0.41	0.57	0.23	0.38
W2V-SBWC	0.17	0.21	0.14	0.34	0.41	0.57	0.21	0.38
BETO	—	—	0.18	0.36	—	—	0.20	0.33

Tabla 4.6: Resultados promedio para 3CosMul (top1 y top5) y Analogy Space Evaluation (Cos y Euc), en test de analogías con Google Analogy.

	Sintáctica							
	Derivación				Inflexión			
Embeddings	top1	top5	Cos	Euc	top1	top5	Cos	Euc
FT-SUC(M)	0.24	0.38	0.26	**0.39**	0.44	0.62	**0.40**	**0.46**
FT-SUC(L)	0.33	0.52	0.19	0.36	**0.57**	**0.75**	0.28	0.40
FT-SUC(NL)	0.32	0.51	0.19	0.36	0.49	0.74	0.29	0.40
FT-SBWC	0.32	0.51	**0.28**	0.36	0.49	0.69	0.26	0.39
FT-Wiki	**0.40**	**0.61**	0.21	0.37	0.41	0.64	0.24	0.38
GV-SBWC	0.09	0.13	0.12	0.34	0.30	0.44	0.17	0.36
W2V-SBWC	0.10	0.17	0.10	0.32	0.38	0.53	0.22	0.38
BETO	—	—	0.16	0.35	—	—	0.13	0.34

Tabla 4.7: Resultados promedio para 3CosMul (top1 y top5) y Analogy Space Evaluation (Cos y Euc), en test de analogías con sección sintáctica de SATS.

	Semántica							
	Enciclopédica				Lexicográfica			
Embeddings	top1	top5	Cos	Euc	top1	top5	Cos	Euc
FT-SUC(M)	0.27	0.45	**0.47**	**0.48**	0.15	0.30	**0.20**	**0.37**
FT-SUC(L)	0.41	0.60	0.37	0.44	0.20	0.36	0.13	0.34
FT-SUC(NL)	0.41	0.60	0.37	0.44	0.20	**0.38**	0.13	0.34
FT-SBWC	0.39	0.58	0.44	0.42	0.18	0.35	0.12	0.34
FT-Wiki	0.32	0.49	0.33	0.42	0.09	0.23	0.12	0.33
GV-SBWC	**0.42**	**0.65**	0.35	0.43	0.16	0.34	0.15	0.35
W2V-SBWC	0.15	0.22	0.28	0.40	**0.21**	0.37	0.13	0.34
BETO	—	—	0.21	0.38	—	—	0.16	0.35

Tabla 4.8: Resultados promedio para 3CosMul (top1 y top5) y Analogy Space Evaluation (Cos y Euc), en test de analogías con sección semántica de SATS.

En un análisis inicial, podemos establecer ciertos aspectos destacables. En primer lugar, los resultados obtenidos en la evaluación a través de analogías de palabras para un word embedding específico, no es constante a través de las distintas clases de analogías. Esto implica que algunos word embeddings representan tipos específicos de relaciones entre palabras. Un ejemplo de lo anterior es el embedding GV-SBWC, el cual tiene resultados muy bajos al ser evaluado con analogías de derivación en SATS, pero obtiene los mejores resultados con analogías enciclopédicas.

Otro aspecto destacable, son los resultados del embedding FT-SUC(M) a través de Analogy Space Evaluation, los cuales son los mejores tanto en Google Analogy como en SATS, y a través de los distintos grupos de analogías. Comparando con los resultados obtenidos utilizando 3CosMul y los resultados del resto de embeddings, podemos notar que los resultados de FT-SUC(M) son muy particulares. A partir de los resultados del resto de embeddings, una posible causa en la diferencia de resultados con Analogy Space Evaluation es la diferencia en dimensiones entre embeddings.

En general, los word embeddings generados a partir del algoritmo FastText consigue los resultados más altos, o cercano a los más altos, a través de los distintos tipos de analogías. Por otro lado, el resto de embeddings se destaca solamente en una o dos clases de analogías.

4.2.3. Resultados para outlier detection

De la misma forma que para las evaluaciones anteriores, el dataset utilizado también fue restringido. Se reemplazaron mayúsculas por minúsculas, y se eliminaron los conceptos que no aparecieran dentro del vocabulario de todos los word embeddings.

Respecto a los resultados, al igual que para la evaluación de similitud semántica, word embeddings generados a partir del algoritmo FastText presenta resultados ligeramente mejores al resto de los embeddings. Lo siguen los embeddings generados por los algoritmos

GloVe, Word2Vec y el modelo BETO. Esto es apreciable a partir de los valores de Acc y OPP obtenidos.

Embeddings	Acc	OPP
FastText-SUC(M)	0.84	0.63
FastText-SUC(L)	**0.86**	**0.66**
FastText-SUC(NL)	0.85	0.65
FastText-SBWC	0.85	0.65
FastText-Wiki	0.84	0.64
GloVe-SBWC	0.82	0.62
W2V-SBWC	0.78	0.53
BETO	0.72	0.44

Tabla 4.9: Resultados de test de outlier detection.

4.2.4. Resultados para cross-match

A diferencia de los otros test de validación, solo se utilizó cross-match para comparar los embeddings FT-SBWC, GV-SBWC y W2V-SBWC, los cuales utilizan el mismo corpus SBWC. Los resultados obtenidos son muy bajos, llegando a ser del orden de 10^{-30}. Esto nos indica que, los distintos modelos de generación de word embeddings utilizan distintas características presentes en el corpus SBWC, al momento de crear embeddings.

	FT-SBWC	GV-SBWC	W2V-SBWC
FT-SBWC	—	7.81×10^{-28}	1.11×10^{-30}
GV-SBWC	7.81×10^{-28}	—	6.7×10^{-28}
W2V-SBWC	1.11×10^{-30}	6.7×10^{-28}	—

Tabla 4.10: Resultados de test de cross-match.

4.3. Evaluación extrínseca

4.3.1. Proceso de evaluación

Para el proceso de clasificación a partir de vectores promedio, se trabajó directamente con el dataset original, pero, los caracteres no alfanuméricos fueron reemplazados por un espacio, se transformó todas las letras mayúsculas a minúsculas y los espacios consecutivos fueron reemplazados por un solo espacio. Puesto que no se necesitaban hacer un manejo especial del dataset, este no fue dividido para ser utilizado en las tareas.

El proceso de clasificación, para la tarea A y B, consistió en la asignación de un vector representante para cada clase y texto a clasificar. Para la tarea A, se clasificaran los argumentos, mientras que, en la tarea B, son los conceptos abiertos los que se clasifican. El vector representante de los textos a clasificar fue el promedio de los vectores de las palabras presentes en el mismo, mientras que, para las clases, el vector representante corresponde al

vector promedio del nombre real de la clase. Determinar a cuál clase pertenece cada texto, se realizó a partir del uso de distancia coseno, es decir, un texto pertenece a la clase que sea más similar, respecto a la distancia coseno de los vectores representantes. Para ambas tareas, el desempeño de los embeddings fue medido a partir de la cantidad de textos correctamente clasificados.

Al igual que con el método de vectores promedios, para el uso de redes neuronales se siguió con el mismo procedimiento de limpieza del dataset, se eliminaron los caracteres no alfanuméricos, las mayúsculas fueron cambiadas por minúsculas y los espacios consecutivos fueron reemplazados por un único espacio, además, fue necesario separar el dataset en dos partes, cada parte utilizada en una tarea distinta.

Para resolver la tarea A, se entrenaron cuatro redes neuronales LSTM, una para cada tópico. El proceso de entrenamiento de las redes neuronales consistió de 200 épocas como máximo, sobre el dataset de entrenamiento del tópico correspondiente. Para optimizar el uso de recursos, y asegurar que la red no sufriera de overfitting, se utilizo el dataset de validación para detener el entrenamiento de forma temprana, en caso de que la red neuronal no mejorará su desempeño después de 10 iteraciones seguidas, el proceso de entrenamiento era detenido tempranamente y se daba comienzo a la evaluación de la red neuronal. Se debe mencionar que, puesto que la cantidad de elementos varía entre clases, durante el proceso de entrenamiento se dio un peso distinto a aquellas clases que contuvieran menos elementos. Producto de lo demandante en recursos que fue el proceso de entrenamiento, este se realizó en Google Collab.

A diferencia de la tarea A, no se entrenó una nueva LSTM para resolver la tarea B, esto es en línea con el proceso de clasificación que se realizó en [15], donde, después de entrenar una red LSTM para resolver la tarea A, se utilizó el mismo modelo para resolver la tarea B, para algún tópico. El proceso de clasificación en esta tarea se separó en dos sub-problemas de clasificación distintos, diferenciados según la información utilizada. Por un lado, realizamos la clasificación de los conceptos abiertos, utilizando solo el texto correspondiente a dichos conceptos. Al mismo tiempo, repetimos el mismo proceso de clasificación, pero, utilizando como información los conceptos abiertos y los argumentos para dichos conceptos.

Para comprobar que los word embeddings tienen una buena representación de las palabras, repetimos una segunda vez todos los procesos de clasificación con redes neuronales, pero los word embeddings fueron entrenados junto a la red neuronal. De esta forma, si se obtienen mejores resultados, eso quiere decir que los embeddings aprendieron nuevas características a partir del dataset.

Cabe mencionar, que durante el desarrollo de los dos métodos de clasificación usados, se omitieron aquellas palabras que no se encontraran dentro del embedding a evaluar, si bien, esto podría afectar los resultados finales, en el peor de los casos, solo el 1 % de las palabras dentro del dataset fueron omitidas, por lo que los resultados no deberían verse afectados.

4.3.2. Resultados y análisis

Respecto a los resultados obtenidos a través del uso de vectores promedio, estos varían entre diferentes embeddings. Para ambas tareas, A y B, los embeddings generados por FastText se destacan por sobre el resto de embeddings.

Una observación importante, es la diferencia entre resultados para la clasificación por vector promedio entre las distintas tareas. En promedio, utilizando este método se obtienen bajos resultados en la tarea A, comparado con los resultados obtenidos en la tarea B. Esto no es una particularidad de algún word embedding, ya que este comportamiento se puede observar a través de todos los word embeddings evaluados (las diferencias entre resultados llegan a ser de 0.4). Es posible que este comportamiento sea debido al dataset utilizado, ya que la cantidad de argumentos abiertos utilizados es diez veces menor que la cantidad de argumentos utilizados en conceptos de gobierno. Otra posibilidad, es debido a que en la tarea B se clasifican los conceptos abiertos, mientras que en la tarea A se clasifican los argumentos para los conceptos de gobierno.

Para la clasificación con redes neuronales, nuevamente se destacan los word embeddings generados a partir de FastText, a través de ambas tareas. Por otro lado, los resultados a partir de entrenar las redes neuronales y word embeddings, notamos que los resultados no son distintos a los obtenidos solo entrenando la red neuronal. Esta situación ocurre en ambas tareas, y ocurre para todos los word embeddings.

También se puede observar que los resultados de las redes neuronales son mejores en la tarea A que en la tarea B. Esto era de esperar, ya que el clasificador a partir de redes neuronales fue entrenado para resolver la tarea A, por lo que era bastante probable que los resultados en la tarea B fueran más bajos.

Finalmente, comparando los resultados obtenidos a través del uso de redes neuronales y los resultados a partir de vectores promedios, notamos que los resultados en las distintas tareas no son iguales. Por un lado, para la tarea A, las redes neuronales obtienen mejores resultados, a través de todos los word embeddings. Mientras que, en la tarea B, es el método de vectores promedios el que alcanza un mejor resultado, a través de todos los embeddings (aunque esta diferencia de resultados es menor que en la tarea A).

4.3.2.1. Resultados tarea A

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	0.13	0.40	0.15	0.46	0.39	0.79	0.15	0.45	0.21	0.53
FastText-SUC(L)	**0.15**	**0.47**	**0.19**	**0.52**	**0.45**	**0.81**	**0.18**	**0.49**	**0.24**	**0.57**
FastText-SUC(NL)	0.12	0.41	**0.19**	0.49	0.41	0.79	0.17	0.43	0.22	0.53
FastText-SBWC	0.12	0.38	**0.19**	0.46	0.42	0.80	0.15	0.41	0.22	0.51
FastText-Wiki	0.11	0.44	0.17	0.44	0.41	0.79	**0.18**	0.45	0.22	0.53
GloVe-SBWC	0.09	0.25	0.08	0.34	0.33	0.75	0.08	0.27	0.15	0.40
W2V-SBWC	0.09	0.26	0.11	0.44	0.34	0.78	0.09	0.32	0.16	0.45
BETO	0.09	0.24	0.06	0.21	0.28	0.71	0.10	0.31	0.13	0.37

Tabla 4.11: Resultados para tarea A, utilizando vectores promedio.

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	**0.51**	**0.81**	**0.56**	**0.83**	**0.67**	**0.92**	**0.55**	**0.84**	**0.57**	**0.85**
FastText-SUC(L)	0.49	0.80	0.54	0.81	0.60	0.90	0.50	0.82	0.53	0.83
FastText-SUC(NL)	0.46	0.78	0.53	0.81	0.60	0.90	0.52	0.81	0.53	0.83
FastText-SBWC	0.49	0.80	0.54	0.82	0.59	0.91	0.54	0.83	0.54	0.84
FastText-Wiki	0.49	0.79	0.54	0.81	0.61	0.91	0.51	0.82	0.54	0.83
GloVe-SBWC	0.48	0.79	0.54	0.81	0.60	0.90	0.51	0.81	0.53	0.83
W2V-SBWC	0.48	0.79	0.54	0.82	0.61	0.90	0.52	0.81	0.53	0.83
BETO	0.41	0.75	0.47	0.76	0.52	0.89	0.45	0.78	0.46	0.80

Tabla 4.12: Resultados para tarea A, utilizando redes neuronales.

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	**0.50**	**0.82**	0.54	**0.83**	**0.67**	**0.92**	**0.55**	**0.83**	**0.57**	**0.85**
FastText-SUC(L)	0.48	0.79	0.53	0.81	0.61	0.90	0.52	0.82	0.54	0.83
FastText-SUC(NL)	0.49	0.80	**0.55**	0.81	0.61	0.90	0.51	0.82	0.54	0.83
FastText-SBWC	0.49	0.80	**0.55**	0.82	0.61	0.91	0.51	0.82	0.54	0.84
FastText-Wiki	0.46	0.78	0.54	0.81	0.61	0.90	0.51	0.81	0.53	0.83
GloVe-SBWC	0.48	0.78	0.53	0.81	0.58	0.90	0.50	0.81	0.52	0.83
W2V-SBWC	0.49	0.79	0.52	0.81	0.61	0.90	0.52	0.81	0.53	0.83
BETO	0.41	0.73	0.48	0.79	0.53	0.90	0.46	0.80	0.47	0.81

Tabla 4.13: Resultados para tarea A, utilizando redes neuronales y entrenamiento de los embeddings.

4.3.2.2. Resultados tarea B

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	0.60	0.84	0.54	0.78	0.61	0.84	0.43	0.82	0.55	0.82
FastText-SUC(L)	0.62	0.87	0.56	**0.79**	**0.63**	**0.85**	0.44	0.84	0.56	**0.84**
FastText-SUC(NL)	0.63	0.88	0.57	**0.79**	0.62	0.82	0.45	0.84	**0.57**	0.83
FastText-SBWC	**0.64**	**0.90**	**0.58**	0.78	0.61	0.83	**0.46**	**0.85**	**0.57**	**0.84**
FastText-Wiki	0.60	0.85	**0.58**	0.78	0.60	0.81	0.45	0.86	0.56	0.83
GloVe-SBWC	0.51	0.75	0.46	0.72	0.51	0.79	0.37	0.74	0.46	0.75
W2V-SBWC	0.59	0.81	0.52	0.71	0.53	0.78	0.42	0.73	0.52	0.76
BETO	0.50	0.73	0.43	0.62	0.47	0.74	0.31	0.63	0.43	0.68

Tabla 4.14: Resultados para tarea B, utilizando vectores promedio.

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	**0.45**	0.75	**0.53**	**0.80**	**0.67**	**0.92**	0.43	0.61	**0.52**	0.77
FastText-SUC(L)	0.42	0.74	0.49	0.79	0.65	0.89	0.40	0.61	0.49	0.76
FastText-SUC(NL)	0.43	**0.79**	0.46	0.79	0.64	0.90	**0.44**	0.66	0.49	**0.79**
FastText-SBWC	0.40	0.75	0.48	0.79	0.60	0.87	0.41	0.60	0.47	0.75
FastText-Wiki	0.38	0.74	0.51	**0.80**	0.61	0.89	0.43	0.65	0.48	0.77
GloVe-SBWC	0.39	0.74	0.51	**0.80**	0.63	0.90	0.41	0.66	0.49	0.78
W2V-SBWC	0.40	0.72	0.49	**0.80**	0.62	0.89	0.42	0.65	0.48	0.77
BETO	0.36	0.71	0.48	0.79	0.65	**0.92**	0.42	**0.71**	0.48	0.78

Tabla 4.15: Resultados para tarea B, clasificación de conceptos abiertos, utilizando redes neuronales.

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	**0.43**	0.75	0.48	0.77	**0.65**	0.87	0.44	0.61	**0.50**	0.75
FastText-SUC(L)	**0.43**	0.76	0.47	0.80	0.60	0.88	0.41	0.66	0.48	0.78
FastText-SUC(NL)	**0.43**	0.75	**0.51**	0.81	0.60	0.89	0.40	0.62	0.49	0.77
FastText-SBWC	**0.43**	0.75	0.49	0.80	0.61	0.88	0.41	0.62	0.49	0.76
FastText-Wiki	0.40	0.74	0.48	0.77	0.63	0.89	0.42	0.63	0.48	0.76
GloVe-SBWC	0.40	0.75	0.48	0.78	0.60	0.88	0.41	0.64	0.47	0.76
W2V-SBWC	0.40	**0.78**	**0.51**	0.81	**0.65**	0.90	0.41	0.60	0.49	0.77
BETO	0.35	0.69	0.48	**0,82**	0.62	**0.94**	**0.46**	**0.70**	0.48	**0.79**

Tabla 4.16: Resultados para tarea B, clasificación de conceptos abiertos, utilizando redes neuronales y entrenamiento de embeddings.

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	**0.41**	0.78	**0.51**	**0.84**	**0.63**	**0.90**	0.36	0.60	**0.48**	0.78
FastText-SUC(L)	0.40	**0.80**	0.49	**0.84**	0.59	0.88	0.35	0.60	0.46	0.78
FastText-SUC(NL)	**0.41**	**0.80**	0.47	0.81	0.56	0.87	0.36	0.61	0.45	0.77
FastText-SBWC	0.39	0.81	0.49	**0.84**	0.56	0.86	0.37	0.63	0.45	0.79
FastText-Wiki	0.39	**0.80**	0.50	**0.84**	0.58	0.88	**0.38**	0.64	0.46	0.79
GloVe-SBWC	0.38	0.77	0.50	0.83	0.58	**0.90**	0.36	0.63	0.46	0.78
W2V-SBWC	0.40	0.79	0.50	0.82	0.56	0.88	0.36	0.66	0.46	0.79
BETO	0.34	0.79	0.46	**0.84**	0.57	0.88	0.36	**0.71**	0.43	**0.80**

Tabla 4.17: Resultados para tarea B, clasificación de argumentos y conceptos abiertos, utilizando redes neuronales.

	Valores		Derechos		Deberes		Institucional		Promedio	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FastText-SUC(M)	0.39	0.79	0.47	0.83	**0.61**	0.88	0.36	0.60	0.46	0.78
FastText-SUC(L)	0.40	0.77	0.48	0.82	0.57	0.89	0.36	0.67	0.45	0.79
FastText-SUC(NL)	0.39	0.80	0.50	0.83	0.58	**0.90**	0.34	0.60	0.45	0.78
FastText-SBWC	0.40	0.80	**0.51**	0.82	0.58	0.89	**0.37**	0.66	0.46	0.79
FastText-Wiki	0.38	**0.81**	0.49	0.83	0.57	0.88	0.34	0.62	0.45	0.79
GloVe-SBWC	0.41	0.80	0.48	0.83	0.58	**0.90**	0.36	0.66	0.46	0.80
W2V-SBWC	**0.43**	0.79	0.48	**0.84**	0.60	**0.90**	**0.37**	0.69	**0.47**	**0.81**
BETO	0.31	0.78	0.45	**0.84**	0.58	**0.90**	**0.37**	**0.73**	0.43	**0.81**

Tabla 4.18: Resultados para tarea B, clasificación de argumentos y conceptos abiertos, utilizando redes neuronales y entrenamiento de embeddings.

Capítulo 5

Conclusión y Trabajo a Futuro

5.1. Conclusión

Finalizando el proceso de evaluación de los distintos word embeddings presentados en este trabajo, podemos concluir que los objetivos, definidos inicialmente, fueron logrados satisfactoriamente. Se logró evaluar y comparar distintos word embeddings para el lenguaje español, como también la creación de una herramienta con la cual evaluar la capacidad representativa de distintos embeddings.

Respecto a la herramienta desarrollada, esta cumple con los objetivos planteados al inicio de este trabajo. Se logró la implementación de distintos métodos de evaluación, cuatro métodos de tipo intrínseco y uno de tipo extrínseco. De los métodos intrínsecos utilizados en la literatura, los implementados en este trabajo de investigación son: similitud semántica, analogías, outlier detection y cross-match. De estos métodos implementados, destacamos la evaluación por similitud semántica y analogías, principalmente debido al gran tamaño de los dataset utilizados, como también por la variedad de métricas que se dispone para las evaluaciones.

Dentro del desarrollo de esta herramienta de evaluación, cabe destacar la construcción de un nuevo dataset para la validación de embeddings a través de analogías. Este nuevo dataset, denominado SATS, expande los tipos de analogías que presenta el dataset Google Analogy, además de reorganizar las clasificaciones de los distintos tipos de analogías. Se espera que este dataset sea un aporte a la validación de word embeddings para el español.

En relación a los distintos word embeddings evaluados, los resultados de la validación intrínseca nos entrega información importante. Una observación general de los resultados, nos dice que los embeddings generados con el algoritmo FastText logran el mejor desempeño en las evaluaciones por similitud semántica, analogías de palabras y outlier detection. A partir de esto, podemos concluir que este algoritmo captura una buena parte de las relaciones que existen entre palabras. Por otro lado, observaciones específicas de los resultados, muestran que el word embedding FT-SUC(L) se destaca por sobre los demás, principalmente en las evaluaciones por similitud semántica y por outlier detection. De entre todos los embeddings, concluimos que FT-SUC(L) es el que representa de mejor forma las relaciones entre palabras.

Por otro lado, el embedding BETO consistentemente obtiene los resultados más bajos, a excepción de ciertos grupos de analogías. Esto nos indica, que utilizar BETO para extraer word embeddings, tal vez no sea la mejor forma de utilizar sus capacidades. Mientras que, los embeddings basados en GloVe y Word2Vec, en general logran resultados similares a los de FastText. La principal diferencia que presentan los word embeddings generados a partir de estos algoritmos, en comparación los generados a partir de FastText, es en la validación por analogías. Analizando los resultados de Word2Vec, nos damos cuenta que no reconoce principalmente las analogías de tipo sintáctico. Mientras que, el word embedding generado por el algoritmo Glove, obtiene resultados similares a Word2Vec, a excepción de las analogías enciclopédicas, en las cuales se destaca por sobre el resto de embeddings. Concluyendo el análisis de estos resultados, podemos decir que, tanto GloVe como Word2Vec, tienen menor capacidad para representar las relaciones entre palabras, a diferencia de FastText.

Con respecto a la evaluación extrínseca realizada en este trabajo, vemos resultados variados. Por otro lado, la clasificación a partir de vectores promedio, se correlaciona con los resultados vistos en la validación intrínseca, manteniendo a los word embeddings basados en FastText como los embeddings con mejores resultados. En cambio, los resultados de la clasificación utilizando redes neuronales, entrega poca información. Una primera observación, es lo cercano que llegan a estar los resultados de los distintos embeddings. Esto es especialmente particular para el caso del embedding BETO, debido a que obtuvo los resultados más bajos en las evaluaciones intrínsecas. Lo anterior nos indica que, si bien BETO no logra resultados destacables en evaluaciones intrínsecas, al ser utilizado en tareas de NLP, llega a tener un desempeño cercano al resto de embeddings.

En general, podemos notar que prácticamente la totalidad de las evaluaciones con redes neuronales, presentan resultados similares a través de los distintos word embeddings, lo cual puede indicar dos cosas. Un primer motivo de esta similitud de resultados, puede ser la red neuronal utilizada. Se puede dar que la red LSTM utilizada en la clasificación, es suficiente para obtener buenos resultados en este problema, por lo que, la información que puedan entregar los embeddings no influencia en gran medida los resultados. Por otra parte, un segundo motivo, pueden ser los mismos embeddings. Es posible que, la información necesaria para obtener buenos resultados en el problema de clasificación, esté representada en todos los embeddings evaluados, con lo cual, se obtienen resultados similares a través de los embeddings. Si bien estas dos hipótesis se pueden corroborar a partir del uso de un embedding generado aleatoriamente, por temas de tiempo no fue posible comprobarlas.

Finalizando este trabajo, cabe recalcar el cumplimiento de los objetivos propuestos. Se logró implementar una serie de métodos de evaluación de word embeddings, los cuales, en conjunto, aportan una herramienta con la cual realizar una serie de evaluaciones estandarizadas. Adicionalmente, se comparó la capacidad representativa de distintos word embeddings para el español, a través de distintas evaluaciones.

5.2. Trabajo a futuro

Respecto al trabajo a futuro, este consiste principalmente en mejorar la herramienta de validación de word embeddings, presentada en este trabajo.

Las posibles mejoras que se podrían realizar se describen a continuación.

Variedad en métodos intrínsecos

Si bien en este trabajo presentamos cuatro métodos de validación intrínseco, tres de los métodos presentados corresponden a métodos de validación intrínseca absoluta (definido en Capítulo 2). Además, estos tres métodos evalúan los word embeddings a través de la capacidad que tienen de representar las relaciones entre palabras, por lo tanto, evalúan características similares.

De lo visto en la literatura, existe una gran cantidad de métodos intrínsecos, de entre los cuales, el uso de tesaurus y grafos de diccionarios, es particularmente interesante. Esto se debe a que son métodos distintos a los presentados en este trabajo y que no necesitan obtenerse directamente del juicio humano, como en el caso de similitud semántica. Es por esto que el desarrollo de dichos métodos, puede ser una posible continuación del trabajo realizado.

Variedad en métodos extrínsecos

En este trabajo se presenta una tarea de clasificación de textos, como un método de validación extrínseco, con el cual podemos determinar la capacidad que tienen los embeddings para ser utilizados en el desarrollo de tareas de NLP. Sin embargo, los resultados de este trabajo no muestran una correlación clara entre la tarea de clasificación de textos y los métodos intrínsecos de evaluación. Lo anterior puede ser debido a que, la tarea extrínseca presentada, no evalúa las mismas características que evalúan los métodos intrínsecos.

Es importante determinar cómo se correlacionan los métodos intrínsecos y extrínsecos, de esta forma, se puede determinar que tipos de word embeddings son más apropiados para distintas tareas de NLP. Debido a esto, una posible dirección en la cual continuar este trabajo, puede ser el desarrollo de nuevos métodos extrínsecos, distintos al presentado en este trabajo, y evaluar la correlación que exista con métodos intrínsecos.

Mejoras en los dataset

De los métodos de validación implementados, tres necesitan de un dataset con el cual realizar las evaluaciones sobre los word embeddings. Si bien se logró obtener un grupo de dataset con los cuales se llevó a cabo las distintas evaluaciones, ciertos aspectos de los dataset podrían mejorarse en un trabajo futuro.

Un primer aspecto a considerar, es la traducción de dataset, principalmente para el método de similitud semántica. Realizar una traducción de los dataset, corre el riesgo de cambiar la relación percibida originalmente entre distintas palabras. Esto es especialmente importante en el caso de similitud semántica, debido a que también se tienen que adaptar los valores de similitud entre palabras. Producto de lo anterior, se debería evaluar la opción de crear un dataset para similitud semántica, construido para el español.

Otro aspecto a considerar, aunque tal vez menos importante, es la cantidad de palabras eliminadas desde los datasets, durante la validación de los distintos embeddings. Si bien hay pocas opciones respecto al caso en que las palabras no se encuentran dentro del vocabulario del embedding, se debería evaluar cómo utilizar conceptos de más de una palabra. Para este trabajo, se omitieron este tipo de conceptos, pero en un trabajo futuro se podría evaluar de qué forma aprovecharlos durante las evaluaciones.

Finalmente, en este trabajo presentamos un nuevo dataset para la validación a través de analogías de palabras, el cual cuenta con analogías de tipo semántica y sintáctica. Sin embargo, este dataset presenta secciones incompletas, específicamente las secciones de semánticas enciclopédica y lexicográfica. Una posible mejora que se puede hacer de este dataset, es la inclusión de nuevos tipos de analogías para dichas secciones.

Bibliografía

[1] Bakarov, A. A survey of word embeddings evaluation methods. *ArXiv abs/1801.09536* (2018).

[2] Barzegar, S., Davis, B., Zarrouk, M., Handschuh, S., and Freitas, A. a. Semr-11: A multi-lingual gold-standard for semantic similarity and relatedness for eleven languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan, May 2018), European Language Resources Association (ELRA).

[3] Blair, P., Merhav, Y., and Barry, J. Automated generation of multilingual clusters for the evaluation of distributed representations. *ArXiv abs/1611.01547* (2017).

[4] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5* (2017), 135–146.

[5] Camacho-Collados, J., and Navigli, R. Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 43–50.

[6] Camacho-Collados, J., Pilehvar, M. T., Collier, N., and Navigli, R. SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (Vancouver, Canada, Aug. 2017), Association for Computational Linguistics, pp. 15–26.

[7] Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. A framework for the construction of monolingual and cross-lingual word similarity datasets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (Beijing, China, July 2015), Association for Computational Linguistics, pp. 1–7.

[8] Cardellino, C. Spanish Billion Words Corpus and Embeddings, August 2019.

[9] Che, X., Ring, N., Raschkowski, W., Yang, H., and Meinel, C. Traversal-free word vector evaluation in analogy space. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (Copenhagen, Denmark, Sept. 2017),

Association for Computational Linguistics, pp. 11–15.

[10] Chiu, B., Korhonen, A., and Pyysalo, S. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 1–6.

[11] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT* (2019).

[12] Drozd, A., Gladkova, A., and Matsuoka, S. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (Osaka, Japan, Dec. 2016), The COLING 2016 Organizing Committee, pp. 3519–3530.

[13] Etcheverry, M., and Wonsever, D. Spanish word vectors from Wikipedia. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (Portorož, Slovenia, May 2016), European Language Resources Association (ELRA), pp. 3681–3685.

[14] Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 30–35.

[15] Fierro, C., Fuentes, C., Pérez, J., and Quezada, M. 200K+ crowdsourced political arguments for a new Chilean constitution. In *Proceedings of the 4th Workshop on Argument Mining* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 1–10.

[16] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web* (New York, NY, USA, 2001), WWW '01, Association for Computing Machinery, p. 406–414.

[17] Firth, J. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford, 1957. reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

[18] Gladkova, A., and Drozd, A. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 36–42.

[19] Gladkova, A., Drozd, A., and Matsuoka, S. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW* (San Diego, California, June 12-17, 2016, 2016), ACL, pp. 47–54.

[20] Gurnani, N. Hypothesis testing based intrinsic evaluation of word embeddings. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 16–20.

[21] Hassan, S., and Mihalcea, R. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Singapore, Aug. 2009), Association for Computational Linguistics, pp. 1192–1201.

[22] Hill, F., Reichart, R., and Korhonen, A. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics 41*, 4 (Dec. 2015), 665–695.

[23] Levy, O., and Goldberg, Y. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning* (Ann Arbor, Michigan, June 2014), Association for Computational Linguistics, pp. 171–180.

[24] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR 2013* (01 2013).

[25] Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Atlanta, Georgia, June 2013), Association for Computational Linguistics, pp. 746–751.

[26] Miller, G. A., and Charles, W. G. Contextual correlates of semantic similarity. *Language and Cognitive Processes 6*, 1 (1991), 1–28.

[27] Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.

[28] Qiu, Y., Li, H., Li, S., Jiang, Y., Hu, R., and Yang, L. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In *CCL* (2018).

[29] Rogers, A., Drozd, A., and Li, B. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)* (Vancouver, Canada, Aug. 2017), Association for Computational Linguistics, pp. 135–148.

[30] Rosenbaum, P. An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society Series B 67* (09 2005), 515–530.

[31] Rubenstein, H., and Goodenough, J. Contextual correlates of synonymy. *Commun. ACM 8* (10 1965), 627–633.

[32] Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), Association for Computational Linguistics, pp. 298–307.

[33] Soares, F., Villegas, M., Gonzalez-Agirre, A., Krallinger, M., and Armengol-Estapé, J. Medical word embeddings for Spanish: Development and evaluation. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop* (Minneapolis, Minnesota, USA, June 2019), Association for Computational Linguistics, pp. 124–133.

[34] Torregrossa, F., Claveau, V., Kooli, N., Gravier, G., and Allesiardo, R. On the correlation of word embedding evaluation metrics. In *Proceedings of The 12th Language Resources and Evaluation Conference* (Marseille, France, May 2020), European Language Resources Association, pp. 4789–4797.

[35] Vulić, I., Baker, S., Ponti, E., Petti, U., Leviant, I., Wing, K., Majewska, O., Bar, E., Malone, M., Poibeau, T., Reichart, R., and Korhonen, A. Multi-simlex: A large-scale evaluation of multilingual and cross-lingual lexical semantic similarity, 03 2020.

[36] Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing 8* (2019), e19.

[37] Zhelezniak, V., Savkov, A., Shen, A., and Hammerla, N. Correlation coefficients and semantic textual similarity. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 951–962.

Apéndices

Apéndice A: Redes neuronales

Redes Neuronales

Las redes neuronales son un tipo de modelo computacional, inspirado en la interacción entre neuronas. Este modelo se compone de neuronas, las cuales se conectan entre sí para transmitir información. Cada una de estas conexiones tienen asignado un peso. Como se puede apreciar en la Figura 5.1, en general, las neuronas dentro de una red, se distribuyen en capas, donde cada neurona de una capa, se conectan a las neuronas de la siguiente capa.

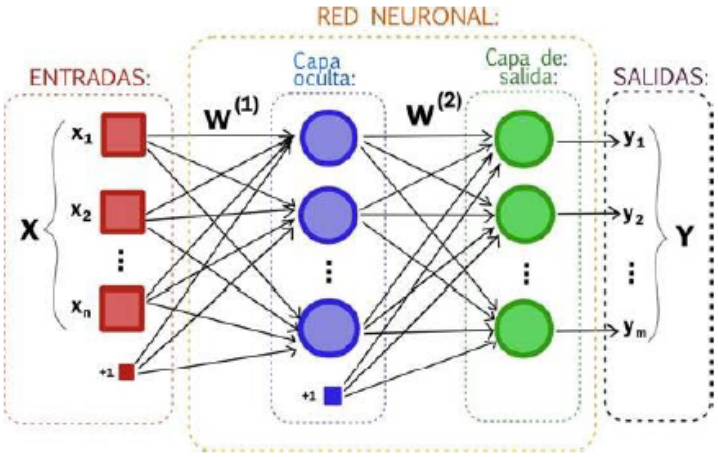

Figura 5.1: Diagrama básico de una red neuronal.

Para utilizar una red neuronal en la resolución de problemas, estas deben pasar por un proceso de entrenamiento. Este proceso de entrenamiento consiste principalmente en el uso de ejemplos resueltos del problema en cuestión. La información inicial de cada ejemplo pasa por cada una de las capas de la red, iniciando por la capa de entrada y terminando en la capa de salida. El paso de información entre capas, involucra los distintos pesos que existen entre cada capa. Finalmente, la información que aparece de la capa de salida es comparada con la solución correcta del ejemplo. Esta comparación entre la información de salida y la solución real del ejemplo, es utilizada para alterar los distintos valores presentes en las conexiones entre neuronas, mejorando la capacidad que tiene la red para entregar la solución correcta. Al finalizar este procedimiento, la red neuronal debería ser capaz de entregar la solución de nuevos ejemplos, a través de la información que aparezca en la última capa de la red.

Redes Neuronales Recurrentes

Las redes recurrentes son un tipo específico de redes neuronales, las cuales pueden ser utilizadas para trabajar con información de tipo secuencial, como videos y texto. La característica más importante de este tipo de red, es su capacidad para "recordar" la información de entrada y de salida de ejemplos anteriores, para luego utilizarlos en el procesamiento de nueva información. La Figura 5.2 describe de forma sencilla cómo está estructurada una red recurrente, al mismo tiempo, también se presenta como se ve una red recurrente al ser utilizada.

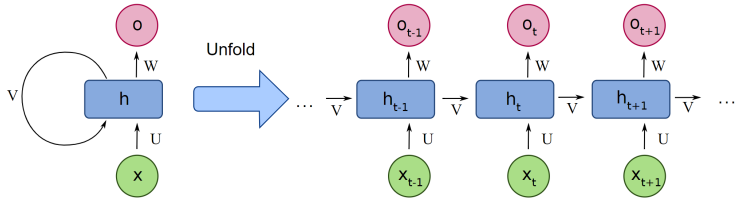

Figura 5.2: Diagrama básico de una red neuronal recurrente. Izquierda: estructura de red recurrente. Derechar: visualizacion del uso de la red.

Si bien, una red recurrente puede recordar información anterior, esta "memoria" no es infinita. Una red recurrente olvida parte de la información, a medida que se procesa una secuencia. Esto implica que, una red recurrente normal, tendrá un mal desempeño procesando secuencias muy largas. Para ello, las redes LSTM (Long Short-Term Memory) eliminan de su memoria la información que ya no es necesaria, al mismo tiempo que actualizan la información importante.

Apéndice B: Ecuaciones

Pearson r

Dado X e Y variables aleatorias, el coeficiente de Pearson se define como:

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{5.1}$$

donde:

$\mathbb{E}$ es la esperanza.

σ_X es la desviación estándar de X.

σ_Y es la desviación estándar de Y.

μ_X es el promedio de X.

μ_Y es el promedio de Y.

Spearman ρ

Dado X e Y variables aleatorias, el coeficiente de Spearman se define como:

$$r_s = \frac{\mathbb{E}[(rg_X - \mu_{rg_X})(rg_Y - \mu_{rg_Y})]}{\sigma_{rg_X} \sigma_{rg_Y}} \tag{5.2}$$

donde:

$\mathbb{E}$ es la esperanza.

rg_X es el ranking de la variable X.

rg_Y es el ranking de la variable Y.

σ_{rg_X} es la desviación estándar de rg_X.

σ_{rg_Y} es la desviación estándar de rg_Y.

μ_{rg_X} es el promedio de rg_X.

μ_{rg_Y} es el promedio de rg_Y.

Kendall τ

Dado X e Y variables aleatorias, sea el conjunto $\{(x_1, y_1), ..., (x_n, y_n)\}$, formado por emparejar uno a uno los elementos de X e Y, de tal forma, que para todo i, x_i es unico en X, y_i es unico en Y. Se dira que el par (x_i, y_i) y (x_j, y_j), con i $< j$, es concordante, si $x_\text{i} < x_j$ e $y_\text{i} < Y_j$, o, $x_\text{i} > x_j$ e $y_\text{i} > Y_j$, en caso contrario, se dira que son discordantes. El coeficiente de Kendall se define como:

$$\tau = \frac{2}{n(n-1)}(C - D) \tag{5.3}$$

donde:

C es la cantidad de pares concordante.

D es la cantidad de pares disconcordante.

Apéndice C: Tablas adicionales

Resultados similitud semántica

	RG-65			MC-30		
Embeddings	r	ρ	τ	r	ρ	τ
FT-SUC(M)	0.85	0.87	0.71	**0.77**	**0.8**	**0.63**
FT-SUC(L)	**0.86**	**0.89**	**0.73**	0.75	0.78	0.61
FT-SUC(NL)	**0.86**	0.88	0.72	0.74	**0.8**	0.62
FT-SBWC	0.81	0.82	0.64	0.7	0.75	0.59
FT-Wiki	0.82	0.88	0.7	0.76	0.79	**0.63**
GV-SBWC	0.74	0.76	0.58	0.5	0.51	0.39
W2V-SBWC	**0.86**	0.85	0.69	0.75	**0.8**	**0.63**
BETO	0.54	0.47	0.33	0.53	0.54	0.4

Tabla 5.1: Resultados para test de similitud semántica con datasets RG-65 y MC-30.

Resultados de analogías de palabras

	AG1		AG2		AG3		AG4		AG5	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FT-SUC(M)	0.7	0.83	0.66	0.81	0.09	0.18	0.11	0.22	0.87	0.97
FT-SUC(L)	0.79	0.87	0.79	0.85	**0.1**	**0.21**	0.22	0.43	**0.96**	**0.997**
FT-SUC(NL)	0.78	0.88	0.77	0.86	0.09	0.2	0.21	0.41	0.95	**0.997**
FT-SBWC	**0.8**	**0.89**	**0.8**	**0.88**	0.09	**0.21**	0.22	0.38	0.83	0.97
FT-Wiki	0.76	0.86	0.78	0.85	0.03	0.07	0.13	0.32	0.84	0.97
GV-SBWC	0.78	**0.89**	0.78	0.86	0.03	0.07	**0.27**	**0.45**	0.82	0.94
W2V-SBWC	0.01	0.02	0.01	0.02	0.03	0.07	0	0.02	0.81	0.92

Resultados test de analogías, para la sección semántica de Google Analogy, utilizando 3CosMul.

	AG6		AG7		AG8		AG9		AG10		AG11		AG12	
Embeddings	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5	top1	top5
FT-SUC(M)	0.19	0.34	0.23	0.34	0.63	0.77	0.79	0.93	0.26	0.35	0.45	0.63	0.31	0.53
FT-SUC(L)	0.28	0.45	0.28	0.45	0.7	0.83	**0.88**	**0.96**	**0.32**	0.47	0.6	0.8	0.37	0.57
FT-SUC(NL)	0.28	0.46	0.3	0.44	**0.75**	**0.84**	**0.88**	**0.96**	0.3	0.43	**0.65**	**0.83**	**0.38**	0.58
FT-SBWC	0.26	0.45	0.3	0.44	**0.75**	0.83	**0.88**	**0.96**	0.3	**0.5**	0.64	0.81	0.36	**0.63**
FT-Wiki	**0.29**	**0.49**	**0.33**	**0.52**	0.68	0.78	0.83	0.93	0.26	0.4	0.61	0.78	**0.38**	0.61
GV-SBWC	0.13	0.27	0.18	0.27	0.65	0.77	0.87	0.94	0.22	0.42	0.45	0.68	**0.38**	0.62
W2V-SBWC	0.22	0.37	0.24	0.33	0.73	0.87	0.39	0.53	0.27	0.46	0.53	0.72	0.47	0.7

Tabla 5.2: Resultados test de analogías, para la sección sintáctica de Google Analogy, utilizando 3CosMul.

	AG1		AG2		AG3		AG4		AG5	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	**0.53**	**0.52**	**0.52**	**0.51**	**0.4**	**0.45**	**0.21**	**0.36**	**0.6**	**0.53**
FT-SUC(L)	0.44	0.47	0.4	0.45	0.32	0.42	0.15	0.34	0.43	0.45
FT-SUC(NL)	0.45	0.48	0.41	0.45	0.31	0.41	0.16	0.35	0.44	0.46
FT-SBWC	0.42	0.46	0.39	0.44	0.3	0.41	0.17	0.35	0.34	0.41
FT-Wiki	0.38	0.44	0.34	0.43	0.25	0.39	0.19	**0.36**	0.35	0.42
GV-SBWC	0.42	0.47	0.42	0.46	0.28	0.4	0.2	**0.36**	0.33	0.41
W2V-SBWC	0.08	0.32	0.05	0.31	0.23	0.38	0.01	0.29	0.34	0.42
BETO	0.25	0.39	0.24	0.38	0.21	0.37	0.15	0.34	0.08	0.32

Tabla 5.3: Resultados test de analogías, para la sección semántica de Google Analogy, utilizando Space Analogy Evaluation.

	AG6		AG7		AG8		AG9		AG10		AG11		AG12	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	**0.25**	**0.38**	**0.21**	**0.37**	**0.49**	**0.5**	**0.71**	**0.62**	**0.36**	**0.44**	**0.28**	**0.41**	**0.42**	**0.46**
FT-SUC(L)	0.17	0.35	0.17	0.35	0.36	0.44	0.59	0.55	0.28	0.4	0.17	0.36	0.27	0.4
FT-SUC(NL)	0.18	0.35	0.17	0.35	0.38	0.44	0.59	0.55	0.29	0.4	0.18	0.36	0.28	0.4
FT-SBWC	0.16	0.34	0.17	0.35	0.35	0.43	0.55	0.52	0.27	0.4	0.13	0.34	0.28	0.4
FT-Wiki	0.17	0.35	0.14	0.34	0.34	0.42	0.52	0.51	0.25	0.38	0.12	0.34	0.29	0.4
GV-SBWC	0.12	0.33	0.18	0.36	0.27	0.4	0.56	0.53	0.18	0.36	0.1	0.33	0.22	0.38
W2V-SBWC	0.14	0.34	0.12	0.33	0.29	0.4	0.28	0.4	0.23	0.38	0.13	0.34	0.26	0.39
BETO	0.1	0.33	0.06	0.31	0.14	0.34	0.25	0.39	0.1	0.33	0.6	0.31	0.12	0.33

Tabla 5.4: Resultados test de analogías, para la sección sintáctica de Google Analogy, utilizando Analogy Space Evaluation.

	D1		D2		D3		D4		D5	
Embeddings	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
FT-SUC(M)	0.08	0.22	0.04	0.1	0.13	0.22	0.19	0.31	0.53	0.71
FT-SUC(L)	0.13	0.38	0.08	0.19	0.23	0.43	0.26	0.43	0.64	0.79
FT-SUC(NL)	0.12	0.36	0.08	0.19	0.21	0.41	0.27	0.43	0.61	0.79
FT-SBWC	0.14	0.43	0.07	0.16	0.22	0.42	0.26	0.44	0.59	0.78
FT-Wiki	**0.31**	**0.64**	**0.11**	**0.28**	**0.32**	**0.58**	**0.37**	**0.62**	**0.71**	**0.9**
GV-SBWC	0.0	0.01	0.01	0.03	0.04	0.08	0.0	0.0	0.18	0.27
W2V-SBWC	0.02	0.04	0.04	0.08	0.09	0.14	0.01	0.02	0.28	0.42
	D6		D7		D8		D9		D10	
Embeddings	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
FT-SUC(M)	0.27	0.46	0.10	0.2	0.18	0.34	0.22	0.38	0.63	0.82
FT-SUC(L)	**0.35**	0.62	**0.17**	**0.33**	0.3	0.54	0.34	0.55	0.77	0.91
FT-SUC(NL)	0.34	0.62	0.15	0.3	0.27	0.53	0.32	0.52	0.78	0.91
FT-SBWC	0.33	**0.63**	0.07	0.2	0.35	0.6	0.34	0.54	0.78	0.92
FT-Wiki	0.31	0.56	0.06	0.16	**0.49**	**0.72**	**0.53**	**0.75**	**0.8**	**0.93**
GV-SBWC	0.04	0.09	0.0	0.01	0.02	0.06	0.03	0.07	0.57	0.72
W2V-SBWC	0.2	0.37	0.04	0.08	0.09	0.17	0.08	0.14	0.18	0.26

Tabla 5.5: Resultados test de analogías, para la sección de derivaciones de CATS, utilizando 3CosMul.

	I1		I2		I3		I4		I5	
Embeddings	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
FT-SUC(M)	0.56	0.71	0.76	0.89	0.55	0.74	0.62	0.82	0.28	0.42
FT-SUC(L)	**0.67**	0.82	**0.9**	**0.97**	**0.67**	**0.83**	0.78	**0.94**	**0.42**	**0.61**
FT-SUC(NL)	**0.67**	0.82	0.88	0.95	0.66	**0.83**	**0.8**	**0.94**	0.4	0.58
FT-SBWC	0.66	**0.83**	0.86	0.93	0.61	0.78	0.53	0.77	0.27	0.48
FT-Wiki	0.59	0.78	0.88	**0.97**	0.59	0.78	0.39	0.75	0.28	0.52
GV-SBWC	0.49	0.73	0.72	0.86	0.42	0.66	0.04	0.09	0.01	0.04
W2V-SBWC	0.59	0.77	0.89	0.96	0.53	0.72	0.14	0.3	0.09	0.16
	I6		I7		I8		I9		I10	
Embeddings	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
FT-SUC(M)	0.36	0.55	0.44	0.62	0.28	0.45	0.3	0.49	0.29	0.47
FT-SUC(L)	**0.51**	**0.71**	**0.62**	**0.82**	**0.44**	**0.65**	0.37	0.58	0.35	0.56
FT-SUC(NL)	0.47	0.67	0.6	0.79	0.43	0.62	0.36	0.61	0.36	0.58
FT-SBWC	0.27	0.45	0.52	0.74	0.4	0.63	**0.45**	**0.68**	0.36	0.61
FT-Wiki	0.18	0.37	0.19	0.44	0.32	0.56	0.35	0.61	0.33	0.59
GV-SBWC	0.06	0.11	0.22	0.34	0.11	0.21	0.43	0.67	**0.39**	**0.64**
W2V-SBWC	0.17	0.27	0.33	0.47	0.21	0.36	0.45	0.66	0.37	0.6

Tabla 5.6: Resultados test de analogías, para la sección de inflexiones de CATS, utilizando 3CosMul.

	E1		E2		E3		E4		E5		E6		E7	
Embeddings	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
FT-SUC(M)	0.4	0.62	0.21	0.4	0.13	0.25	0.15	0.3	0.57	0.7	0.0	0.0	0.13	0.22
FT-SUC(L)	0.54	0.77	0.29	0.52	0.27	0.46	0.28	0.47	**0.69**	**0.79**	0.0	0.0	0.16	0.38
FT-SUC(NL)	0.56	0.78	0.27	0.5	0.27	0.46	0.28	0.48	0.68	0.78	0.0	0.0	**0.19**	0.38
FT-SBWC	0.58	0.77	0.25	0.47	0.26	0.45	0.3	0.46	0.57	0.75	0.0	0.02	**0.19**	0.38
FT-Wiki	0.52	0.71	0.16	0.31	0.13	0.25	0.19	0.39	0.59	0.78	0.0	0.0	0.14	0.22
GV-SBWC	**0.64**	**0.81**	**0.33**	**0.62**	**0.29**	**0.54**	**0.37**	**0.59**	0.49	0.67	**0.07**	**0.14**	0.18	**0.42**
W2V-SBWC	0.0	0.0	0.07	0.15	0.03	0.07	0.06	0.11	0.6	0.76	0.0	0.0	0.0	0.0

Tabla 5.7: Resultados test de analogías, para la sección enciclopédica de CATS, utilizando 3CosMul.

	L1		L2	
Embeddings	Top1	Top5	Top1	Top5
FT-SUC(M)	0.22	0.45	0.08	0.15
FT-SUC(L)	**0.28**	0.51	0.11	0.21
FT-SUC(NL)	0.27	**0.52**	**0.13**	0.23
FT-SBWC	0.24	0.47	0.12	0.23
FT-Wiki	0.08	0.26	0.09	0.19
GV-SBWC	0.2	0.44	0.12	**0.24**
W2V-SBWC	**0.28**	0.49	**0.13**	**0.24**

Tabla 5.8: Resultados test de analogías, para la sección de lexicográfica de CATS, utilizando 3CosMul.

	D1		D2		D3		D4		D5	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	0.1	0.32	**0.08**	**0.31**	**0.1**	**0.32**	**0.35**	**0.42**	**0.4**	**0.45**
FT-SUC(L)	0.08	0.32	0.05	0.3	0.09	0.31	0.25	0.38	0.3	0.4
FT-SUC(NL)	0.08	0.32	0.05	0.3	0.09	**0.32**	0.25	0.38	0.3	0.4
FT-SBWC	0.08	0.32	0.05	0.3	0.08	**0.32**	0.25	0.38	0.28	0.39
FT-Wiki	**0.13**	**0.33**	0.06	**0.31**	0.09	**0.32**	0.31	0.41	0.34	0.42
GV-SBWC	0.06	0.31	0.05	**0.31**	0.08	**0.32**	0.11	0.33	0.1	0.33
W2V-SBWC	0.03	0.3	0.03	0.3	0.05	0.3	0.12	0.34	0.12	0.33
BETO	0.11	**0.33**	0.04	**0.31**	0.07	0.31	0.21	0.37	0.22	0.37
	D6		D7		D8		D9		D10	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	0.24	0.37	0.15	0.34	**0.23**	**0.37**	**0.35**	**0.43**	**0.57**	**0.54**
FT-SUC(L)	0.18	0.35	0.11	0.33	0.16	0.34	0.26	0.39	0.46	0.48
FT-SUC(NL)	0.18	0.35	0.11	0.33	0.15	0.34	0.24	0.38	0.46	0.48
FT-SBWC	0.15	0.34	0.1	0.32	0.15	0.34	0.25	0.38	0.41	0.46
FT-Wiki	0.18	0.35	0.1	0.33	0.19	0.35	0.31	0.41	0.42	0.46
GV-SBWC	0.26	0.39	**0.16**	**0.35**	0.1	0.33	0.09	0.32	0.4	0.45
W2V-SBWC	0.13	0.34	0.09	0.32	0.07	0.31	0.13	0.34	0.2	0.36
BETO	**0.32**	**0.42**	0.14	0.34	0.11	0.33	0.18	0.36	0.18	0.35

Tabla 5.9: Resultados test de analogías, para la sección de derivaciones de CATS, utilizando Space Analogy Evaluation.

	I1		I2		I3		I4		I5	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	**0.5**	**0.5**	**0.59**	**0.55**	**0.51**	**0.51**	**0.58**	**0.54**	**0.28**	**0.39**
FT-SUC(L)	0.39	0.45	0.45	0.48	0.35	0.43	0.43	0.46	0.23	0.37
FT-SUC(NL)	0.4	0.45	0.46	0.48	0.35	0.43	0.44	0.47	0.23	0.37
FT-SBWC	0.39	0.45	0.45	0.48	0.33	0.42	0.32	0.41	0.18	0.35
FT-Wiki	0.36	0.43	0.46	0.48	0.36	0.43	0.18	0.35	0.16	0.34
GV-SBWC	0.3	0.41	0.38	0.44	0.26	0.39	0.09	0.32	0.08	0.32
W2V-SBWC	0.33	0.42	0.38	0.44	0.3	0.41	0.26	0.39	0.11	0.33
BETO	0.16	0.35	0.17	0.35	0.14	0.34	0.16	0.35	0.19	0.35
	I6		I7		I8		I9		I10	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	**0.33**	**0.42**	**0.44**	**0.47**	**0.28**	**0.4**	**0.25**	**0.39**	**0.24**	**0.39**
FT-SUC(L)	0.25	0.38	0.31	0.41	0.2	0.37	0.14	0.34	0.13	0.34
FT-SUC(NL)	0.24	0.38	0.31	0.41	0.21	0.37	0.14	0.34	0.13	0.34
FT-SBWC	0.17	0.35	0.31	0.41	0.21	0.37	0.12	0.34	0.11	0.33
FT-Wiki	0.11	0.33	0.27	0.39	0.21	0.37	0.13	0.34	0.13	0.34
GV-SBWC	0.06	0.31	0.16	0.35	0.18	0.36	0.11	0.33	0.1	0.33
W2V-SBWC	0.13	0.33	0.26	0.39	0.23	0.38	0.12	0.34	0.11	0.33
BETO	0.03	0.3	0.15	0.34	0.08	0.32	0.1	0.33	0.08	0.32

Tabla 5.10: Resultados test de analogías, para la sección de inflexiones de CATS, utilizando Space Analogy Evaluation.

	E1		E2		E3		E4		E5		E6		E7	
Embeddings	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc	Cos	Euc
FT-SUC(M)	**0.58**	**0.54**	**0.54**	**0.52**	**0.38**	**0.44**	**0.38**	**0.44**	**0.46**	**0.47**	**-0.35**	0.16	**0.2**	0.36
FT-SUC(L)	0.44	0.47	0.45	0.48	0.32	0.42	0.3	0.41	0.32	0.41	**-0.35**	0.16	0.16	0.34
FT-SUC(NL)	0.45	0.47	0.44	0.47	0.32	0.42	0.3	0.41	0.32	0.41	**-0.35**	0.16	0.17	0.35
FT-SBWC	0.43	0.46	0.41	0.46	0.28	0.4	0.27	0.4	0.25	0.38	-0.36	0.16	0.17	0.35
FT-Wiki	0.39	0.45	0.39	0.45	0.28	0.4	0.29	0.4	0.28	0.39	-0.35	0.16	**0.2**	**0.37**
GV-SBWC	0.45	0.48	0.43	0.47	0.33	0.42	0.28	0.4	0.25	0.38	-0.33	**0.17**	0.15	0.35
W2V-SBWC	0.06	0.31	0.33	0.42	0.36	**0.44**	0.36	**0.44**	0.27	0.39	-0.36	0.16	0.02	0.3
BETO	0.26	0.39	0.31	0.42	0.23	0.38	0.21	0.38	0.06	0.31	-0.38	0.15	0.13	0.34

Tabla 5.11: Resultados test de analogías, para la sección enciclopédica de CATS, utilizando Space Analogy Evaluation.

	L1		L2	
Embeddings	Cos	Euc	Cos	Euc
FT-SUC(M)	**0.27**	**0.39**	**0.13**	**0.34**
FT-SUC(L)	0.17	0.35	0.09	0.32
FT-SUC(NL)	0.17	0.35	0.08	0.32
FT-SBWC	0.15	0.35	0.09	0.32
FT-Wiki	0.15	0.34	0.08	0.32
GV-SBWC	0.19	0.36	0.1	0.33
W2V-SBWC	0.17	0.35	0.08	0.32
BETO	0.23	0.38	0.09	0.33

Tabla 5.12: Resultados test de analogías, para la sección de lexicográfica de CATS, utilizando Space Analogy Evaluation.