Capítulo 1

Introducción

El tratamiento de lenguaje natural (PLN) por parte de computadoras ha sido siempre un punto de gran interés de la comunidad. Desde los trabajos de John McCarthy y Marvin Minsky en los años 1950 hasta la actualidad, se han publicado miles de trabajos al respecto. Históricamente han habido dos paradigmas para el tratamiento del lenguaje: uno basado en reglas y otro en aprendizaje automático.

El primer enfoque busca construir reglas que capturen el conocimiento de expertos en lingüística para resolver una problemática particular, construyendo reglas que generalicen lo suficiente pero sin dejar de lado los casos particulares. Mientras que ambos enfoques se desarrollaron paralelamente, no fue hasta la adopción masiva de Internet, junto con los inmensos volúmenes de texto que consecuentemente se generaron, que los métodos de aprendizaje automático comenzaron a ser significativemente más exitosos.

Todas las tareas en el área de PLN involucran, en algún nivel, trabajar con palabras. Puesto que los algoritmos de aprendizaje automático comúnmente utilizados como soluciones de dichas tareas se alimentan de vectores, es usual requerir de un mecanismo para traducir de unas a otros.

La forma más simple de realizar esta traducción es posiblemente asociarle, a cada palabra, un vector bajo un esquema *one-hot*: esto es, a cada palabra se le asigna un vector donde todas las componentes son cero salvo por un uno en la correspondiente a la palabra. Sin embargo, dado que todos los vectores resultan equidistantes, esta representación no tiene la capacidad de generalizar bien: las palabras *el* y *lenguaje* están a la misma distancia que *el* y *de*, por lo que se pierde información de su función sintáctica y semántica. Por esta razón, la búsqueda de métodos que asignen representaciones parecidas a palabras relacionadas es un área de investigación de particular interés.

Basándose en la hipótesis distribucional de Harris [21], la cual plantea que palabras que ocurren en contextos similares tienen significados similares, se han desarrollado históricamente una gran cantidad de métodos para realizar la tarea de traducción, principalmente basados en construir una matriz de coocurrencias la cual asocia, a cada par de palabras del vocabulario, una medida de similitud obtenida a partir de los contextos en los cuales

éstas aparecen juntas. Estos métodos, donde entran técnicas como el análisis semántico latente (LSA), logran resultados relativamente satisfactorios.

Sin embargo, recientemente han surgido una nueva generación de métodos para representar palabras que, mediante el uso de técnicas inspiradas en el modelado de lenguaje mediante redes neuronales, han llegado resultados sorprendentes para la comunidad del PLN. Estos algoritmos se basan en métodos de aprendizaje automático no supervisado que, a partir de grandes volúmenes de texto, consiguen representaciones vectoriales con una fuerte capacidad de almacenar información sintáctica y semántica. Entre los modelos más conocidos de este grupo se encuentran los propuestos por Mikolov et al. [36, 37, 38], *Skipgram* y *CBOW*, comúnmente referidos bajo el nombre `word2vec`.

Una de las grandes atracciones de `word2vec` es la capacidad de los modelos generados de expresar relaciones sintácticas y semánticas mediante operaciones algebraicas. Quizás el ejemplo más conocido de este fenómeno es la relación que cumplen los vectores de las palabras *reina*, *rey*, *mujer*, y *hombre*, donde el primer vector puede ser aproximadamente recuperado de los siguientes mediante la relación $reina \approx rey - hombre + mujer$. A partir del surgimiento de `word2vec`, se ha reinvigorado la investigación en el área, y se han publicado numerosos artículos estudiando los modelos y proponiendo variantes de los mismos.

1.1. Objetivos del Proyecto

En este marco es que surge el presente proyecto de grado, el cual plantea, mediante el estudio en profundidad de las técnicas existentes para la representación vectorial de palabras, los siguientes objetivos:

- La generación de un corpus de texto para el español del orden de dos mil millones de palabras. Dado que los algoritmos anteriormente mencionados requieren de grandes cantidades de texto, es necesario como primer paso del proyecto -y de cualquier investigación en el área-, generar un corpus de tales características. Para la obtención del mismo se deberá realizar crawling masivo en la web, pues no existen corpus semejantes para el español de forma libre.
- La creación de una herramienta que permita, de una manera simple y con una interfaz accesible, explorar el corpus generado de manera interactiva, pudiendo realizar búsquedas sobre el mismo. La herramienta también brindará acceso a un servidor de cómputo especializado que permitirá entrenar algoritmos de modelado de manera remota, pudiendo almacenar allí todos los datos y descargarlos en la medida que sea necesario. Ofrecerá a su vez la posibilidad de evaluar la calidad de los modelos masivamente sobre diferentes conjuntos de pruebas, y permitirá la visualización de esta información fácilmente.
- La construcción de un conjunto de datos de evaluación que permitan analizar la calidad de las representaciones logradas, mediante la traducción de los principales conjuntos de pruebas utilizados en la literatura en inglés, para poder formar así

una idea de cómo se diferencia el comportamiento de los modelos en el idioma español. También se crearán nuevos datos de prueba especializados en el lenguaje de estudio que tengan en cuenta las particularidades del mismo, sobretodo en el ámbito sintáctico.

- La comparación de la calidad de las representaciones obtenidas de los distintos modelos de representaciones para el español, estudiando así la existencia de particularidades de un lenguaje sobre otro que influencien el comportamiento de los modelos entrenados.
- Dejar disponible una infraestructura que ayude a futuras investigaciones en el área a no partir de cero, dejando como fruto los resultados de los anteriores puntos mencionados. De particular interés es brindar acceso a modelos vectoriales previamente entrenados, así como dejar un corpus que permita obtener nuevas representaciones que sean competitivas con el estado del arte. Con la herramienta creada, se pretende facilitar el entrenamiento de vectores especializados para distintas tareas del PLN.

1.2. Estructura del Informe

El presente informe se estructura de la siguiente forma: en el capítulo 2 se realiza un relevamiento del estado del arte para la construcción de representaciones vectoriales de palabras, dando una clasificación de los métodos existentes junto a una breve descripción de los mismos. Se busca que al finalizar este capítulo el lector conozca las técnicas principales y qué resultados éstas obtienen.

El capítulo 3 describe el proceso de construcción del corpus de texto, y se realiza un relevamiento de la literatura en construcción de corpus para el español. Se describen las técnicas empleadas, la infraestructura utilizada, y los problemas enfrentados. Se detalla la composición del corpus generado, junto a un breve análisis de su calidad.

En el capítulo 4 se presenta la herramienta construida, con sus objetivos iniciales y decisiones de diseño. Se describen sus funcionalidades y se procede a detallar los principales aspectos de su implementación.

El capítulo 5 se exponen los resultados de las evaluaciones realizadas. Se comienza por presentar la metodología empleada y los conjuntos de prueba construidos, y finalmente se realiza una comparación con el estado del arte en otros idiomas.

Por último, el capítulo 6 plantea las conclusiones y propuestas para trabajo futuro.

Capítulo 2

Representación Vectorial de Palabras

En esta sección se pretende realizar una recorrida por la literatura en el área de representación vectorial de palabras, un tema que se ha ubicado en los últimos años en el centro de la escena en muchas tareas de PLN. Se dará un pantallazo de qué entendemos por representaciones vectoriales, de dónde surgen, y cuáles son los principales trabajos asociados a su desarrollo, mediante un tratamiento más bien superficial que permita poner en perspectiva el trabajo que se realiza en el presente proyecto de grado.

2.1. Conceptos Básicos

En los últimos años ha habido un importante cambio de paradigma en PLN, donde la mayor investigación, y el mayor éxito, se ha obtenido con la aplicación de métodos de aprendizaje automático para el tratamiento del lenguaje, en oposición a métodos basados en reglas.

Los métodos de aprendizaje automático son técnicas principalmente estadísticas que manipulan vectores, por lo general reales. Dado que la unidad semántica básica del lenguaje natural, las palabras[1], son entidades discretas y de gran cardinalidad, es necesario realizar un mapeo entre estos dos espacios. Por esta razón es que existe un gran interés en buscar y caracterizar mecanismos que permitan construir representaciones vectoriales para las palabras del lenguaje.

Formalmente, si V es el vocabulario con el que se está tratando en un contexto dado (por ejemplo, $V = \{\text{a}, \text{ábaco}, \text{abajo}, \ldots\}$), decimos que una función $F : V \to \mathbb{R}^N$ induce una representación vectorial de dimensión N para dicho vocabulario si le asigna un vector de $\mathbb{R}^N$ a cada palabra de V (e.g. $F(\text{a}) = (1{,}1, 0{,}9, \ldots, 0{,}2)$, $F(\text{ábaco}) = (2{,}3, 0{,}1, \ldots, 0{,}1)$).

[1]Se podrían considerar entidades más pequeñas como los morfemas como unidad básica, pero el punto se mantiene.

La construcción de funciones F que tengan buenas propiedades es un objeto de estudio muy interesante y por lo tanto muy tratado, pues la performance de una gran cantidad de algoritmos de PLN dependen directamente de la calidad de las mismas. Dependiendo el contexto en que se esté trabajando, puede ser deseable requerirle a F tener una algunas propiedades particulares:

- Puesto que los algoritmos de aprendizaje automático por lo general trabajan con vectores reales continuos, suele ser un punto positivo que palabras que están relacionadas o que tienen significados similares tengan asociados vectores que también sean similares en cierto nivel, pudiendo explotar así la continuidad de los métodos utilizados.

  Por ejemplo, imaginemos que se quiere entrenar un clasificador de tópico trivial utilizando *Support Vector Machines* para asociar a cada palabra del vocabulario una etiqueta indicando el tema del que trata (supongamos *Deportes* y *No Deportes*). Será altamente deseable que palabras que suelen aparecer en los mismos contextos estén cerca en el espacio vectorial resultante, asociando vectores más cercanos entre sí a palabras como *tenis*, *fútbol* y *gol* que a palabras como *gato* o *comida*. Así, el algoritmo logrará separar las palabras pertenecientes a ambas categorías más efectivamente.

- Uno de los principales problemas de trabajar con el lenguaje natural es la dimensionalidad de los vocabularios con los que se trata. Por esta razón, otro requerimiento interesante para las representaciones vectoriales es que asocien a las palabras vectores de $\mathbb{R}^N$ donde $N \ll |V|$, logrando así hacer tratables las dimensiones de las entradas.

- Otra alternativa al punto anterior que también hace tratables entradas de dimensiones altas es utilizar representaciones vectoriales dispersas: esto es, que los vectores asociados a las palabras tengan casi todas sus entradas en cero. Esto permite el empleo de técnicas de tratamiento de matrices dispersas, mejorando así la eficiencia desde el punto de vista computacional y de almacenamiento.

- Dado que se está embebiendo un espacio muy rico en estructura, tanto desde el punto de vista semántico como sintáctico, es de particular interés lograr mantener dicha estructura en el espacio de destino: esto es, obtener una representación que exhiba regularidades lingüísticas en alguna medida.

  Por ejemplo, sería satisfactorio que la relación que existe entre el par de palabras *correr* y *corriendo* (esto es, el gerundio) puediera ser capturada matemáticamente, y que fuera el mismo que presentan las palabras *jugar* y *jugando*. O, desde un punto de vista semántico, que el par de palabras *frío* y *calor* presente las mismas propiedades que *alto* y *bajo*.

Desde las distintas áreas del PLN se han desarrollado una gran cantidad de métodos para la construcción de representaciones vectoriales que cumplan con algunas de las propiedades anteriores, donde el elemento común suele ser la utilización de grandes cantidades de texto a partir del cuál inferir buenas representaciones. Por esta razón, las

representaciones vectoriales suelen recibir diversos nombres, dependiendo de la literatura consultada.

Uno de los enfoques más tradicionales recibe el nombre de *modelos semánticos distribucionales* (o DSMs por su sigla en inglés) los cuales, inspirados en el campo de la semántica estadística y la hipótesis distribucional [21], hacen uso de estadísticas globales derivadas de un corpus de texto para aprender representaciones que capturen la semántica de las palabras. En este grupo de técnicas se puede encontrar el modelo LSA [15, 13], junto con sus principales variantes, y el modelo HAL [33, 32].

El otro gran enfoque en la construcción de representaciones vectoriales es el proveniente de la comunidad de modelado de lenguaje mediante redes neuronales. Estos son métodos predictivos que utilizan información local para construir modelos de lenguaje a partir de redes neuronales [3] y redes neuronales recurrentes [43]. Una de las salidas de estos algoritmos son vectores para las palabras del lenguaje, las cuales suelen denominarse *vectores o modelos neuronales*, o *embeddings* de palabras.

Otros nombres utilizados incluyen *representaciones vectoriales continuas*, *espacios de palabras semánticos*, o simplemente *vectores de palabras*.

Destacamos también la existencia de técnicas basadas en métodos aglomerativos para la construcción de representaciones vectoriales más compactas. Un ejemplo son los métodos basados en el algoritmo de Brown [5] (denominados *Brown Clusters*): bajo este esquema, se realiza un *clustering* jerárquico de palabras utilizando estadísticas de los bigramas del corpus, tales como su información mutua, y se construye un código que busca minimizar el largo de las representaciones para grupos de palabras semánticamente relacionadas. También existen diferentes técnicas que se basan en modelos ocultos de Markov [30] o en K-means [31]. Sin embargo, dado que estos modelos no han tenido los mejores resultados en la literatura, y que tienen la desventaja de no producir representaciones continuas, limitaremos nuestro estudio en el presente proyecto a los dos enfoques anteriormente presentados.

Vectores de palabras obtenidos con distintos mecanismos han sido utilizados exitosamente para una gran cantidad de tareas de PLN, en especial en los últimos años, donde se han superado resultados del estado del arte mediante la combinación de los mismos con aprendizaje profundo (o *deep learning*, como es conocido en la literatura). Mencionamos a continuación algunos de los principales resultados.

- En [54, 55], se utilizan vectores basados en modelos neuronales para realizar análisis de sentimiento en críticas de películas a nivel de árboles de parsing, superando levemente los resultados obtenidos anteriormente.
- En [11, 12], los autores plantean un nuevo esquema para resolver problemas de PLN utilizando redes neuronales convolucionales cuya única entrada son vectores de palabras, logrando resultados muy cercanos al estado del arte en tareas como *POS tagging* (etiquetado de categorías gramaticales), etiquetado de roles semánticos, y reconocimiento de entidades con nombre.

- En [57], se mejoran los resultados obtenidos en reconocimiento de entidades con nombre y en *chunking* al agregar vectores de palabras como *features* en un algoritmo de CRF (campos aleatorios condicionales).
- En [56], mediante la utilización de representaciones vectoriales y de *autoencoders*, se mejora el estado del arte en materia de detección de parafraseo en textos cortos.
- En [35], el autor entrena una red neuronal recursiva utilizando como única entrada vectores de palabras para generar un modelo de lenguaje para el checo, que posteriormente es utilizado para reconocimiento automático de habla.
- En [60], se generan vectores de palabras bilingües, donde se aprenden representaciones para dos lenguajes de manera simultánea, obteniendo buenos resultados en traducción automática de frases cortas.

En las siguientes secciones presentaremos una descripción más detalladas de los principales enfoques para la construcción de representaciones vectoriales, llegando así a los métodos del estado del arte en el área.

2.2. Modelos Estadísticos

Los modelos estadísticos se originan en el área de la lingüística computacional, basándose en estudios teóricos de semántica distribucional de lingüistas como Zellig Harris, con su *hipótesis distribucional* [21], y John Firth, con su noción de *contexto de situación* [18], haciendo referencia a la dependencia que tiene el significado de las palabras del contexto en el que ocurren (o, como lo plantea Firth, *Conocerás una palabra por la compañía que mantiene*). El hecho de que surjan hace tantos años dotan al área de una literatura muy rica y extensa. A continuación daremos una breve reseña, recorriendo los resultados más significativos.

La idea principal detrás de estos modelos es, pues, describir a las palabras del vocabulario según el contexto en el que estas ocurren en un determinado corpus de texto utilizado para la construcción de las representaciones. Con este fin, se construye un vector $v \in \mathbb{R}^N$ que captura, de alguna manera, información acerca del contexto donde aparece la palabra.

Qué se considera el contexto de una palabra es una decisión propia de cada método; para un elemento lingüístico particular, se podría considerar la pertenencia a una región de texto dado (e.g. la pertenencia a una oración, un párrafo, o hasta un documento entero), o se puede considerar definiciones que sean independientes de la estructura, considerando la relación con otros elementos lingüísticos (e.g. la cantidad de *coocurrencias* con otra palabra).

Para el punto anterior, también es importante definir qué *métrica de asociación* se estará midiendo respecto al contexto elegido: el caso más básico es contar la cantidad de

ocurrencias de la palabra dicho contexto, pero existen técnicas que capturan mejor esta relación.

Otro punto importante a considerar es que la cantidad de contextos que se consideran puede ser muy elevada: en el caso de coocurrencias con elementos lingüísticos, el vector asociado a un contexto tendrá un tamaño del orden de $|V|$ elementos (donde V es el vocabulario), mientras que si se consideran regiones de texto se estará tratando con dimensiones aun mayores. Por esta razón, es necesario evaluar el empleo de técnicas de reducción de dimensionalidad, o al menos métricas de asociación que generen representaciones dispersas que hagan al resultado tratable computacionalmente (esto es, vectores con muchos ceros).

Por último, cómo utilizar la representación final obtenida y cómo medir la similitud entre dos palabras es un tema no menor a resolver, que dependerá de la aplicación final de los vectores. Por ejemplo, cuando se trata con representaciones dispersas, medir la similitud con una distancia euclidea tendrá resultados inferiores que la distancia coseno, pues esta última estará menos sesgada a la cantidad de componentes nulas que hay en el vector.

Resumiendo, a la hora de construir un modelo estadístico para la representación vectorial de palabras, es necesario tomar las siguientes decisiones:

- **Tipo de contexto**: utilizar regiones de texto o relaciones con otros elementos lingüísticos.
- **Forma que tendrá este contexto**: la forma que tendrán las regiones de texto consideradas o la ventana de coocurrencia con los elementos lingüísticos.
- **Métrica de asociación**: cómo se medirá la relación de una palabra con su contexto.
- **Reducción de dimensionalidad**: cómo se manejará el tamaño elevado de las representaciones obtenidas.
- **Integración**: cómo utilizar los vectores en la tarea en la que se trabaja. Este punto no es particular a los modelos estadísticos, pero es bueno resaltarlo, pues es de suma importancia. Podría involucrar, por ejemplo, determinar cómo medir la similitud entre dos palabras si se van a utilizar directamente o, si servirán como *features* en un algoritmo de aprendizaje automático, cómo encajarlos en esta maquinaria más grande.

Siguiendo los puntos anteriores, un modelo estadístico se podría formalizar de la siguiente manera:

- Un vocabulario V, conjunto de palabras que se considerarán. V podría formarse las palabras que están presentes en el corpus, por ejemplo.

- Un conjunto de contextos C, que captura las decisiones respecto al tipo y forma de los contextos a utilizar. C podría ser una lista de documentos (donde $c_1 \in C$ corresponde al primer documento, $c_2 \in C$ al segundo, etc.), o un conjunto de elementos lingüísticos con los que se medirá la coocurrencia (podrían ser las palabras de V, por ejemplo).

- Una matriz de asociación $A \in \mathcal{M}_{|V|\times|C|}(\mathbb{R})$, que captura la métrica de asociación entre el vocabulario y el contexto. Si $f : V \times C \to \mathbb{R}$ es la métrica de asociación entre la palabra y su contexto, entonces $A = (a_{ij}) = f(v_i, c_j)$.

- Una transformación de reducción de dimensionalidad $R : \mathcal{M}_{|V|\times|C|}(\mathbb{R}) \to \mathcal{M}_{|V|\times d}(\mathbb{R})$, donde d es la dimensión de los vectores resultantes asociados a las palabras de V. R podría eventualmente ser la identidad (y $d = |C|$), o podría ser la aplicación de técnicas como SVD.

- La matriz de traducción T, resultado de aplicarle R a la matriz de asociación A (i.e. $T = R(A)$), es la matriz utilizada para ir de una palabra codificada bajo un esquema *one-hot* al espacio de vectores asociado al vocabulario V: esto es, las filas de la matriz corresponden a los vectores de las palabras de V.

Cabe aclarar que la formalización recién presentado es más bien de carácter general y de ningún modo exhaustiva: mientras que la mayoría de de los modelos estadísticos siguen el esquema anterior, hay variantes que no encajan a la perfección pero no por eso se los deja de considerar modelos estadísticos.

Como se mencionó anteriormente, los modelos estadísticos tienen una larga historia, principalmente en las áreas de recuperación de información (IR, por sus siglas en inglés) y representaciones vectoriales semánticas.

Uno de los primeros trabajos en el área fue el modelo de análisis latente semántico (LSA) propuesto en [15], también llamado indizado latente semántico (o LSI, en especial en el área de IR). Siguiendo un modelo basado en resultados de la psicología, los autores proponen utilizar documentos enteros como contextos, y contar la cantidad de veces que dos palabras dadas coocurren como métrica de asociación: esto es, la matriz de asociación contará la cantidad de veces que dos palabras aparecen en un mismo documento. Una vez construida esta matriz, los autores proponen utilizar una descomposición en valores singulares (SVD, por sus siglas en inglés) de dicha matriz, truncándola a los d valores singulares más grande, donde d varía dependiendo del caso (los autores utilizan 100 en algunos casos, 200 en otro).

En [13], los autores continúan con la técnica anterior, pero variando la métrica de asociación: consiguen un mejor resultado utilizando TF-IDF en lugar de las frecuencias absoultas para las coocurrencias. El objetivo principal de LSI es la recuperación de documentos basados en una consulta, donde una palabra dada se proyecta al espacio de dimensión reducida y se devuelven los documentos más cercanos en dicho espacio.

En [26] los autores profundizan y formalizan los anteriores modelos, incluso planteando LSA como un modelo de adquisición del lenguaje en humanos. Además proponen la uti-

lización la entropía (de teoría de la información) como métrica de asociación alternativa, mejorando los resultados obtenidos en tareas de similitud de palabras.

Continuando con estas ideas, [24] propone una variante al algoritmo anterior, denominado LSI probabilístico (o pLSI, pLSA). Plantea formalmente la noción de variables latentes, denominadas tópicos, relacionadas a documentos y palabras. Bajo este esquema, en lugar de utilizar SVD para reducir la dimensionalidad, los autores plantean fijar a priori la cantidad de tópicos, que corresponde a la dimensión d resultante de los vectores, y luego expresar cada documento como una combinación (denominada *mixture*) de dichos tópicos. Cada palabra tiene una cierta probabilidad de utilizarse en un tópico dado, y todos los parámetros se ajustan mediante un algoritmo de maximización de la esperanza.

Otra variante al último algoritmo fue la propuesta en [4], donde los autores plantean un modelo generativo denominado LDA (asignación de Dirichlet latente) que sigue un esquema de pLSI donde la distribución de fondo para los tópicos se asume que sigue una distribución de Dirichlet.

Los anteriores modelos emplearon contextos basados en documentos. Otra corriente paralela fue propuesta en [33, 32], denominada análogo en el hiperespacio para el lenguaje (o HAL). En esta propuesta, los contextos se forman en base a las coocurrencias de las palabras del vocabulario en una ventana de largo fijo dado. Esto es, se construye una matriz de tamaño $|V| \times |V|$, y en la entrada (v_i, v_j) se acumula el resultado la métrica de asociación $f(v_i, v_j)$ para cada coocurrencia de las palabras v_i y v_j en el corpus.

En el planteo original, los autores proponen formar una matriz de $|V| \times 2|V|$, y consideran una ventana de largo 10 a la izquierda y otra a la derecha de la palabra central (v_i, en el caso anterior). La métrica de asociación utilizada es el inverso de la distancia entre ambas palabras (donde la palabra más cercana toma el valor 1, mientras que el más lejano 1/10).

Para la reducción de la dimensionalidad, los autores se quedan con las 200 componentes con más varianza, bajo la observación que para la mayoría de las palabras la varianza es casi nula. Esto corresponde a quedarse con las 200 palabras que presentan mayor varianza en sus entradas. En [59] se obtienen mejores resultados (aunque a un costo computacional mayor) realizando un análisis de componentes independientes (ICA) para reducir la dimensionalidad.

Distintas alternativas se han empleado también para la reducción de dimensionalidad, además de SVD, ICA o las técnicas probabilísticas de pLSA. Una de las que mayor éxito ha tenido fue el indizado aleatorio (también conocido como *random indexing*, *random mapping*, o *random projection*), propuesto en [25], el cual propone quedarse con d dimensiones elegidas aleatoriamente de la matriz de coocurrencias. Esta técnica se basa en el lema de Johnson-Linderstrauss, que plantea que las distancias entre los vectores de palabras entre el espacio original y el reducido se preservarán casi completamente, siempre y cuando el valor de d sea suficientemente grande. En [53], se emplea esta técnica en lugar de SVD siguiendo un esquema HAL.

En cuanto a métricas de asociación, mucho trabajo se ha centrado alrededor de la

información mutua puntual entre dos palabras (PMI). Esta medida fue propuesta inicialmente en [10] y busca modelar correctamente nociones de asociatividad entre palabras del lenguaje: el hecho de que frases como *cuesta arriba* son más comunes que *cuesta derecha*. El trabajo inicial de los autores no estuvo relacionado a representaciones vectoriales, pero fue tomado más adelante por [58] con ese fin. En dicho trabajo se presenta un algoritmo que usa PMI como medida de asociatividad, y compara los resultados con LSA.

De todos modos, no es hasta [7] que se utiliza dicha métrica para la construcción explícita de modelos vectoriales del lenguaje. En este trabajo los autores hacen una evaluación sistemática de distintos parámetros para la construcción de vectores basándose en un esquema HAL. Una de los principales aportes fue la introducción de la PMI positiva (PPMI), igual a la PMI pero donde los valores negativos se sustituyen por cero. La intuición detrás de este cambio, plantean los autores, es que los valores negativos implican que el par de palabras tiene menos coocurrencias de lo esperado, punto que se podría dar por, por ejemplo, un corpus de tamaño insuficiente o ruidoso. Otra ventaja de esta métrica es que la matriz de asociación resultantes es dispersa, lo que ayuda a que su cálculo sea manejable computacionalmente y a que la reducción de dimensionalidad no sea imprescindible.

Además de métricas de asociación, los autores realizan pruebas con la dimensionalidad de los vectores resultantes (quedándose con las d palabras de mayor frecuencia únicamente), con el tamaño del corpus utilizado para tomar las estadísticas, y con el tamaño de la ventana empleada. Llegan finalmente a que, en todos los casos, los mejores resultados se obtienen con la métrica PPMI, en especial cuando los tamaños de las ventanas son muy chicos (1 ó 2 a ambos lados de la palabra). En cuanto al corpus y la dimensionalidad, cuánto mayor mejor, aunque está claro que esto afectará muy negativamente a la eficiencia computacional.

Esta línea de investigación se continúa en [8], donde se evalúa también la utilización de técnicas de reducción de dimensionalidad. Los autores muestran que los resultados se pueden mejorar significativamente aplicando SVD y truncando a los d valores singulares más grandes (donde d varía según el problema objetivo, pero suele estar cerca de $d = 1000$). Además, logran mejorar aún más los resultados de mediante una variante de SVD propuesta por [9], donde la matriz de valores singulares se eleva a un exponente P, nivelando dichos valores para quitarles peso a los más grandes.

El estado del arte en modelos estadísticos sigue principalmente el esquema recién planteado, pero introduciendo algunas variantes adicionales inspiradas en los modelos neuronales: en [29], los autores utilizan distintas heurísticas para preprocesar el corpus con el que se entrena el modelo (como realizar un *subsampling* de palabras muy frecuentes, normalizar los vectores resultantes, o eliminar palabras raras del vocabulario), logrando así mejorar significativamente la calidad de las representaciones obtenidas.

2.3. Modelos Neuronales

La representación distribuida de conceptos mediante el uso de redes neuronales fue presentada inicialmente en [23]. En este artículo, el autor plantea la idea de usar los pesos de las capas internas de redes neuronales básicas como la representación de un concepto particular en un espacio continuo. Esta idea fue también propuesta por Rumelhart en [52], donde se realiza una exposición más completa del mecanismo.

Aunque Hinton y Rumelhart plantearon inicialmente la idea de aprender conceptos abstractos (como relaciones familiares), la técnica no tardó en extenderse al lenguaje natural. En 1990, Elman [16] aplica redes neuronales recursivas básicas (denominadas *Elman networks*, utilizadas para realizar modelado a través del tiempo) a la tarea de predicción de oraciones y utiliza los pesos internos aprendidos por la red como representación de las palabras. En las pruebas que llega a realizar el autor (rudimentarias, debido al poder de cómputo disponible en la época), consigue resultados prometedores, donde la representación de palabras como *girl* y *woman* quedan más próximas que palabras como *cat* y *mouse*.

De todos modos, debido a la dificultad de entrenar dichas redes neuronales, la técnica quedó en desuso por varios años más.

No fue hasta 2003 donde Bengio presenta en [3] una alternativa competitiva al modelado de lenguaje a través de ngramas[2] mediante la utilización de redes neuronales. El autor plantea la idea de utilizar el espacio real para modelar el lenguaje, con el fin de explotar la continuidad local para obtener una mejor generalización, cualidad que carecen los modelos estadísticos basados en espacios discretos. Por ejemplo, en un modelo de ngramas, la ocurrencia de la frase "el gato corre" es completamente independiente (y no elevará la probabilidad) de la frase "el perro corre". Esto implica, además que es necesario una mayor cantidad de parámetros para la estimación: una probabilidad de ocurrencia por ngrama. Otro punto positivo es que, al requerir menos parámetros, es posible utilizar contextos más grandes: mientras que el estado del arte en modelos estadísticos utilizan 3 ó 4 palabras, los modelos continuos llegan a utilizar hasta 10 ó 15.

La idea principal detrás modelado continuo de lenguaje es distribuir la densidad de probabilidad de la siguiente palabra (dada una secuencia de palabras inicial) de forma más inteligente: mientras que los modelos estadísticos la distribuyen uniformemente alrededor de cada combinación de ngramas (e.g. de igual manera para "el gato árbol" que para "el gato camina", dada la frase "el gato corre"), los modelos continuos buscan que palabras relacionadas reciban más masa de probabilidad que las que no están relacionadas.

En su artículo, Bengio presenta un esquema distinto al usual para entrenar el modelo de lenguaje: plantea una arquitectura que consta de vectores continuos de palabras y una función paramétrica definida sobre dicho espacio de vectores, el cual modela la probabilidad de la siguiente palabra dada una ventana de N palabras precedentes. El autor

[2]Los modelos de ngramas, o modelos estadísticos, buscan caracterizar al lenguaje en base a la probabilidad de secuencias cortas de palabras denominadas ngramas, y fueron el estado del arte hasta la década pasada.

utiliza una red neuronal para modelar la probabildad, pero deja abierta la posibilidad de usar modelos como modelos de combinación gaussiana (o *gaussian mixture models*). Bajo esta arquitectura, tanto los vectores de palabras como la función de probabilidad son aprendidas en simultáneo, utilizando un corpus texto de gran tamaño como entrada al algoritmo.

La principal innovación de este artículo fue la separación de los vectores de palabras y la función de modelado de la probabilidad; de hecho, en [34] ya se habían utilizado redes neuronales para el modelado de lenguaje. El esquema de Bengio logra así mejorar significativamente el estado del arte en materia de modelado de lenguaje, en especial a lo que respecta a la eficiencia (comparando con otros modelos basados en redes neuronales), desatando una nueva ola de investigación en el área, tanto en cuanto al modelado mediante redes neuronales como a la utilización de los vectores de palabras que deja como resultado adicional el algoritmo.

A partir de este artículo, surgen también variantes de la arquitectura anteriormente descrita. En [35, 39] Mikolov presenta una arquitectura alternativa, donde se aprende primero por separado las representaciones vectoriales utilizando un método no supervisado basado en recorrer bigramas en un corpus de texto; esto es, utiliza una ventana de largo dos, si se sigue el esquema anterior, pero los aprende independientemente del modelo de lenguaje. Luego estos vectores son utilizados para entrenar una red neuronal que modela el lenguaje. Bengio en su artículo original presentó una variante similar a esta, donde propone incluso la idea de utilizar vectores de LSA fijos, en lugar de entrenarlos junto a la red, pero obtuvo resultados inferiores.

Otra variante propuesta por Mikolov [43, 42, 41, 40] fue, en lugar de utilizar una ventana de largo fijo y una red neuronal estándar, utilizar una red neuronal recursiva simple para modelar el lenguaje. Esta arquitectura es similar a las *Elman networks*, pero el autor tiene mejores resultados por utilizar técnicas de entrenamiento modernas que atenúan los problemas usuales de entrenar redes neuronales recursivas (RNNs). Como derivado de este modelo también se generan representaciones vectoriales. El autor incluso ofrece una herramienta de código abierto (*RNN toolkit*) para la creación de modelos de lenguaje que sigan esta arquitectura, pudiendo obtener tanto los vectores como el modelo de lenguaje [40].

Por otro lado, la aparición de estos modelos generó gran interés en el uso de representaciones distribucionales para distintas tareas de PLN, más allá del modelado de lenguaje. En [11], los autores presentan una nueva arquitectura genérica para la resolución de problemas de PLN utilizando exclusivamente aprendizaje profundo (esto es, sin ingeniería de features), inspirados por el esquema propuesto por Bengio. Esta propuesta se basa en la noción de aprendizaje por transferencia (conocido como *transfer learning* o *multi-task learning*), donde se entrena un modelo para resolver más de una tarea a la vez, con el objetivo de que el conocimiento que adquiere en una tarea pueda ser de utilidad en otra.

Con este fin, entrenan primero un modelo de lenguaje siguiendo la arquitectura de Bengio y luego, con los vectores resultantes, entrenan en simultáneo cuatro redes neuronales para distintas tareas (POS tagging, etiquetado de roles semánticos, etiquetado de entidades con nombre, detección de sinónimos), propagando los errores hasta los vectores.

Logran así mejorar el estado del arte en todas las tareas que prueban, donde destacan particularmente los resultados de SRL por considerarla la tarea más compleja. El resultado de este trabajo es de gran importancia, porque plantea la utilización de representaciones vectoriales como una nueva alternativa para la resolución de problemas y muestra que es una técnica competitiva con las técnicas existentes.

Con este nuevo auge de aplicaciones de las representaciones, se comienzan también a buscar mejorar la eficiencia en la generación de vectores de palabras independientemente del modelado de lenguaje. En [44], los autores proponen variantes a la arquitectura de Bengio que buscan ser mejores desde el punto de vista computacional. De las cuatro variantes que proponen, una de ellas, un modelo *log-bilinear* (LBL), consigue incluso mejores resultados en la tarea de modelado de lenguaje. Esta técnica es luego extendida a una versión jerárquica y más rápida denominada HLBL en [45], haciendo uso de una versión jerárquica de la función *softmax*, propuesta por Bengio en [47].

En [36], Mikolov muestra que los vectores generados por una RNN (en particular, por una RNN entrenada utilizando su *RNN toolkit*) presentan regularidades lingüísticas muy interesantes: además de los vectores ser muy buenos en tareas de similitud y relación entre palabras[3], éstos logran capturar relaciones sintácticas y semánticas a través de vectores específicos a cada relación. El autor ejemplifica este fenómeno a través de la resolución de analogías[4], y construye un conjunto de pruebas compuesto por analogías sintácticas.

Utilizando dicho conjunto, compara el rendimiento en esta tarea con sus vectores, con vectores basados en modelos estadísticos (aunque no hace una comparación exhaustiva, por lo que no obtiene buenos resultados), con los vectores generados por Collobert y Weston [11], y por los generados por Mnih y Hinton (HLBL [45]), obteniendo los mejores resultados con los propios y los HLBL.

Los resultados obtenidos en el anterior artículo motivaron la propuesta, por parte de Mikolov en [37], de dos arquitecturas novedosas para la construcción de vectores, centrada en la tarea de analogías y en mejorar la eficiencia computacional. En estas arquitecturas se deja de lado la RNN y el modelado de lenguaje, y se centra exclusivamente en la construcción de vectores.

Ambos esquemas propuestos se basan en definir una ventana de largo fijo, simétrica alrededor de una palabra central, y plantear un problema de optimización basado en predecir la palabra central dado el contexto o vice versa. El primer caso recibe el nombre de *Continuous Bag-of-words* (o CBOW, esquematizado en la [figura 1]), mientras que el segundo recibe el nombre de *Skipgram* (o SG, esquematizado en la [figura 2]).

La probabilidad se modela en los dos casos utilizando exclusivamente un softmax jerárquico, como el propuesto por Morin y Bengio en [47], con el fin de mejorar la eficiencia. Estos dos nuevos modelos logran, por lo tanto, mejorar los resultados en la tarea

[3]Por ejemplo, decidir si una palabra es sustituible por otra en un contexto dado. En las siguientes secciones se dará una descripción más detallada de estas tareas.

[4]Por ejemplo, vectores que capturen la relación de género entre dos palabras, de modo que a partir de los vectores de *hombre*, *mujer* y *rey* se pueda recuperar la palabra *reina*. Más adelante se entrará en mayor detalle.

de analogías a un costo computacional significativamente menor que el de entrenar una RNN completa.

Cabe notar que, mientras que el autor lo plantea como la utilización de una red neuronal de una única capa, también se puede ver directamente como una regresión logística multinomial, por lo que el modelo se está simplificando enormemente con la finalidad de mejorar la eficiencia computacional, en especial cuando se lo compara con modelos basados en RNNs. Dado el *trade-off* que existe entre la complejidad de los modelos estadísticos y la cantidad de datos que éstos pueden procesar, este punto ubica a la propuesta de Mikolov como un modelo simple que requiere de muchos datos. De hecho, en las pruebas que realiza el autor se utilizan corpus de texto del orden de los miles de millones de palabras y, cuanto más aumenta el tamaño del mismo, mejores resultados obtiene.

En [38], Mikolov cierra su trabajo en representaciones vectoriales de palabras presentando extensiones sobre el modelo Skipgram. El artículo comienza formalizando el modelo: se presenta la función objetivo, que previamente había obviado, y se detalla la utilización del softmax jerárquico. Luego se presenta una alternativa a esta última técnica que logra mejorar significativamente los resultados, basada en *Noise-contrastive Estimation* (NCE), propuesta inicialmente por Gutmann y Hyyvärinen [20] y aplicada por Mnih y Teh para modelado de lenguaje [46]. Esta técnica, que denomina *negative sampling* (NS), se basa en generar ejemplos negativos de uso del lenguaje: esto es, además de utilizar el texto proveniente del corpus de entrenamiento, se genera texto aleatorio, bajo la premisa de que será inválido gramaticalmente, como ejemplo de mal uso del lenguaje.

El autor también presenta una serie de heurísticas para el procesamiento del corpus, como la realización de *subsampling* de palabras muy frecuentes (i.e. ignora aleatoriamente palabras que son demasiado comunes en el corpus) y la eliminación de palabras muy raras, que mejoran aún más los resultados.

Por último, junto a la publicación de este artículo, Mikolov presenta un nuevo conjunto de pruebas mucho más extenso, que abarca tanto casos sintácticos como semánticos. También hace pública su implementación de los modelos CBOW y Skipgram, bajo una herramienta denominada `word2vec`. Este punto no es de menor importancia, porque contribuyó a aumentar el interés en el tema, en especial entre el público amateur, y permitió reproducir y comparar los resultados con distintos métodos de manera más correcta desde un punto de vista metodológico.

2.4. Modelos Híbridos

Los dos tipos de representaciones vectoriales descritos tienen mucha literatura detrás, pero al haber surgido independientemente, carecían de comparaciones sistemáticas y completas entre ellos. La primera de estas evaluaciones se realiza en [2].

En este artículo, los autores reúnen catorce conjuntos de pruebas utilizados por la

comunidad para los problemas de similitud entre palabras, analogías, y otros. Someten a estas pruebas a modelos estadísticos (modelos basados en contar, como los llama el autor) y a modelos neuronales (modelos basados en predecir). Para los primeros utiliza vectores construidos con la herramienta `DISSECT` [14], basados principalmente en esquemas PMI con reducción de dimensionalidad con SVD. Para los segunndos utiliza la herramienta provista por Mikolov en [38], `word2vec`.

Los resultados que obtienen los autores presentan a los modelos neuronales como grandes ganadores, donde obtienen mejores resultados en todas las pruebas realizadas. Esta conclusión lleva a la comunidad a investigar qué es lo que hace mejores a los métodos neuronales por sobre los estadísticos.

Siguiendo esta línea de pensamiento, en [29] el autor busca identificar qué es lo que hace que Skipgram con Negative Sampling (SGNS) funcione tanto mejor que un modelo PPMI que utiliza SVD. Los resultados a los que lleva, sin embargo, son contradictorios con los de Baroni. Plantea que la diferencia entre la performance de ambos métodos se debe a que SGNS tiene una ventaja por utilizar, además del modelo básico, una serie de heurísticas en el preprocesamiento del corpus y el posprocesammiento de los vectores que mejoran drásticamente los resultados.

De esta forma, Levy identifica una serie de nueve heurísticas de uno y otro modelo, que los considerará hiperparámetros, y los adapata para ambos esquemas. Además de hacer esto, entrena todos los vectores utilizando exactamente el mismo corpus de datos (punto que no hizo Baroni, pues utilizó vectores pre-entrenados descargados de Internet) y compara contra el mismo conjunto de pruebas. Así, utilizando una metodología más robusta que en el estudio anterior, llega a que los resultados de los modelos estadísticos y los modelos neuronales son prácticamente equivalentes, con los primeros con una leve ventaja. De todos modos, el autor resalta que SGNS es mucho más eficiente computacionalmente, lo que le permite utilizar más datos.

Más allá de las comparaciones, los dos enfoques anteriores no son necesariamente ortogonales. En [28] se muestra que SGNS está en realidad siguiendo un esquema muy similar a los enfoques estadísticos, donde se realiza una factorización implícita de una matriz SSPMI: esto es, una matriz PPMI donde la medida de asociación es $f(v_i, v_j) = \text{máx}(PMI(v_i, v_j) - \log(k), 0)$, con k un hiperparámetro del modelo. Este resultado es muy importante, pues conecta directamente dos enfoques históricamente independientes. El autor plantea que la ventaja que SGNS tiene sobre la matriz PPMI estándar se da en que su factorización está ponderada de modo de no dar demasiada importancia a las palabras más comunes, una de las debilidades principales de la métrica PPMI.

Siguiendo estos resultados, han surgido diversos métodos que plantean un esquema híbrido, donde se busca hacer explícita la tarea de aprendizaje que se realiza, basándose en las lecciones aprendidas de los enfoques ya presentados. Uno de estos métodos es GloVe, presentado en [49], donde los autores construyen una matriz A, similar a la matriz de asociación de los modelos estadísticos, donde la función de asociación busca explícitamente quitar peso a las palabras más frecuentes y no sobrerrepresentar a las muy poco frecuentes. Luego plantea una reducción de dimensionalidad basada en factorizar dicha matriz a través de un método iterativo en lugar de usar SVD, lo que lo hace compu-

tacionalmente más eficiente. Los resultados que obtienen los autores son superiores a los obtenidos por SGNS en el conjunto de pruebas provisto por Mikolov[5].

Los modelos estadísticos, neuronales e híbridos han resultado bastante similares en cuanto a los resultados obtenidos. Reconociendo este punto, la comunidad se está centrando en modelos híbridos que aprovechen las lecciones aprendidas por los tres enfoques, buscando nuevas métricas de asociación (definiendo matrices de asociación explícitamente) y nuevas formas de factorizarlas (ya sea con variantes de SVD, o con métodos iterativos).

2.5. Evaluación y Estado del Arte

Hasta ahora se presentaron distintos enfoques para la construcción de representaciones vectoriales, pero no se ha detallado qué entendemos por estado del arte: esto es, cuándo consideramos que una representación es mejor que otra, y cómo las comparamos.

Es posible comparar las representaciones de forma implícita y explícita. La primera implica evaluar los cambios en la performance en un algoritmo de aprendizaje automático donde se usan; esto es, como parte de una solución de un problema de PLN más grande. La segunda refiere a evaluar directamente la calidad de los vectores obtenidos mediante alguna tarea de PLN desarrollada específicamente para el caso.

Para la evaluación explícita, el objetivo es diseñar un experimento cuyo resultado esté, en la medida de lo posible, correlacionado a la calidad de una solución de PLN de mayor porte cuando se utilizan estas representaciones. Esto permite optimizar una métrica más definida y más fácil de calcular. Por ejemplo, si se está construyendo una solución para el procesamiento automático del habla (ASR), y se supiera que representaciones que funcionan mejor en la tarea de analogías mejoran el resultado final, sería mucho más fácil y eficiente probar distintos hiperparámetros y modelos evaluando con esa métrica, que entrenar todo un modelo de ASR de principio a fin para evaluar si los vectores produjeron mejores resultados[6]. En especial porque en un modelo complejo, pueden haber varios componentes que afecten la calidad final de la solución, por lo que no se sabrá si fueron los vectores los que mejoraron los resultados o no.

De todos modos, esta correlación entre la evaluación explícita e implícita de representaciones vectoriales por lo general no se puede probar, y es una suposición que se toma cuando se realiza una evaluación explícita.

En la literatura se han usado muchas tareas para la evaluación explícita de vectores, algunas de las cuales ya han sido mencionadas brevemente. Una de las más antiguas es la tarea de similitud y relación entre palabras, que busca decidir si una palabra es sustituible por otra en un contexto dado. Para esto, se construye un conjunto de pruebas compuesto por pares de palabras y un puntaje, determinado por un grupo de humanos, de qué tan

[5]Cabe notar que en [29] no se logra reproducir este resultado, aunque aun así es un método muy competitivo.

[6]Esto es particularmente problemático para los vectores de palabras, pues suelen ser el primer componente en una solución PLN.

similares son dichas palabras. El objetivo es obtener un buen nivel de correlación entre los puntajes de los pares de palabras y las distancias[7] en el espacio de vectores, medida utilizando la correlación de Spearman. Distinguimos también entre similitud, donde se mide relaciones más fuertes, como la sinonimia y la hiponimia, de la relación (*relatedness* en la literatura), donde se incluyen relaciónes más amplias, posiblemente temáticas.

Existen muchos conjuntos de prueba para esta tarea. Uno de los más utilizados es WordSim353 [17], compuesta de 353 pares de palabras, diferenciados entre relación y similitud. También muy populares están MEN [6], compuesto de 1000 pares de palabras, SimLex999 [22], y Mechanical Turk [51].

Otra alternativa que ha surgido recientemente para el estudio de regularidades lingüísticas de representaciones vectoriales es la tarea de analogías, propuesta inicialmente por Mikolov en [36]. En este escenario, se cuenta con dos pares de palabras que mantienen una misma relación (sintáctica o semántica) y se busca determinar la cuarta palabra a partir de las anteriores tres. Por ejemplo, si se cuenta con los pares de palabras *correr* y *corriendo*, *jugar* y *jugando*, el objetivo es, a partir de los vectores de *correr*, *corriendo* y *jugar*, lograr recuperar la cuarta palabra, *jugando*.

Para esto Mikolov originalmente propone en [36] utilizar la función 3CosAdd para recuperar la cuarta palabra, definida como:

$$\arg \max_{b' \in V} \cos(b', b - a + a')$$

Donde los pares de analogías son (a, a') y (b, b'). Sin embargo, en [27], Levy prueba que una mejor función para la recuperación de analogías es 3CosMul, definida como:

$$\arg \max_{b' \in V} \frac{\cos(b', b) \cos(b', a')}{\cos(b', a) + \epsilon}$$

Donde ϵ es un valor muy pequeño (e.g. $\epsilon = 0{,}001$) utilizado para evitar la división entre cero.

Mikolov introdujo inicialmente un conjunto de 8000 analogías exclusivamente sintácticas en [36], con relaciones como plurales y conjugaciones verbales. Luego introduce en [37] un conjunto de analogías más amplio que cuenta con cerca de 20000 analogías, tanto sintácticas como semánticas.

Existen también otras tareas que se han utilizado, aunque en menor medida, para la evaluación explícita de vectores, como la utilización del TOEFL (*Test of English as Foreign Language*, una prueba utilizada para medir el conocimiento de inglés) de [26], la categorización de conceptos de [1], y la preferencia de selección de [48].

En cuanto a evaluación implícita de las representaciones vectoriales, y cómo se relacio-

[7] Es usual utilizar la distancia euclidea o la distancia coseno.

na con los resultados de la evaluación explícita, no existe mucho trabajo en la literatura. Un trabajo preliminar que toca superficialmente este punto es [50], donde los autores utilizan distintos modelos vectoriales (SGNS, GloVe, los vectores de Collobert y Weston, y Brown Clusters) como features de algoritmos de clasificación secuencial (principalmente CRFs, para resolver los problemas de POS tagging y NER). Llegan a que, al usar distintas representaciones bajo este esquema (donde los vectores son una feature más), los resultados mejoran de manera muy similar para los distintos modelos: esto es, se obtiene la misma ganancia utilizando Brown Clusters (que genera una representación más rudimentaria) que SGNS.

Existe también un poco de escepticismo por parte de algunos investigadores en el área respecto a qué beneficios proveen los vectores de palabras. Edward Grefenstette escribe [19] que las representaciones vectoriales parecen ser principalmente una forma de aprendizaje por transferencia, donde se entrena con la tarea de analogías o similitud entre palabras para resolver una tarea más compleja. Plantea que esto es beneficioso cuando se trabaja con corpus de entrenamiento demasiado chicos, pues ayuda a la generalización, pero que no son necesarios e incluso pueden perjudicar la performance cuando se tienen suficientes datos.

Otro caso que fortalece este argumento es el hecho que en [11] se obtienen muy buenos resultados en el etiquetado de roles semánticos, pero los vectores son inferiores en la tarea de analogías (como muestra [36]). Esto, de todos modos, puede resultar del hecho que los autores ajustan los valores de los vectores en la medida que entrenan con las otras tareas.

Estos puntos son importantes y requieren de mayor investigación, pues determinan qué tanto es necesario buscar nuevos y mejores modelos vectoriales. Es posible, por ejemplo, que pequeñas diferencias en los resultados de evaluación explícita sean compensados por el modelo que los usa, como pasa con los Brown Clusters en [50]. En este caso, se podría incluso utilizar un corpus más chico para entrenar las representaciones, o vectores más simples.

De todos modos, en el presente proyecto se estará realizando una evaluación puramente explícita de los modelos vectoriales entrenados, por varias razones. En primer lugar, el objetivo es realizar una investigación del comportamiento de representaciones vectoriales en general, no de un problema de PLN particular. Segundo, porque no hay un estándar en evaluación implícita con el que se pueda comparar los resultados, ni que sean una aplicación directa de vectores de palabras. Tercero, porque realizar una evaluación implícita sería mucho más costoso computacional y metodológicamente, lo que nos impediría probar con una gran variedad de modelos. Y por último, porque las tareas con las que se estarán evaluando, analogías y similitud de palabras, entre otros, tienen de por sí aplicaciones directas que son de gran utilidad.

En cuanto a los modelos vectoriales que se evaluarán, se decidió optar por un representante de cada enfoque: Skipgram y CBOW como modelos neuronales, una matriz PPMI con SVD como modelo estadístico, y GloVe como modelo híbrido. Se eligen estos cuatro modelos por ser los que consiguen los mejores resultados en las tareas de analogías y similitud de palabras en la literatura en inglés, evitando así sesgarnos a un esquema de representación vectorial particular, y pudiendo evaluar si alguno de ellos tiene un

comportamiento particular para el idioma español.

Bibliografía

[1] Abdulrahman Almuhareb. "Attributes in Lexical Acquisition". En: *PhD Thesis, University of Essex* (2006).

[2] Marco Baroni, Georgiana Dinu y German Kruszewski. "Don't count, predict! A systematic comparison of context-counting vs . context-predicting semantic vectors". En: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.* (2014), págs. 238-247. DOI: 10.3115/v1/P14-1023.

[3] Yoshua Bengio y col. "A Neural Probabilistic Language Model". En: *The Journal of Machine Learning Research* 3 (2003), págs. 1137-1155. ISSN: 15324435. DOI: 10.1162/153244303322533223. arXiv: arXiv:1301.3781v3.

[4] David M Blei, Andrew Y Ng y Michael I Jordan. "Latent Dirichlet Allocation". En: *Journal of Machine Learning Research* 3.4-5 (2012), págs. 993-1022. ISSN: 15324435. DOI: 10.1162/jmlr.2003.3.4-5.993. arXiv: 1111.6189v1.

[5] Peter Brown y col. "Class-based n-gram models of natural language". En: *Comput. Linguist.* 18.4 (1992), págs. 467-479.

[6] Elia Bruni y Daniel Gatica-perez. "Multimodal distributional semantics Marco Baroni , Thesis Advisor". En: 48.December (2013).

[7] John a Bullinaria y Joseph P Levy. "Extracting semantic representations from word co-occurrence statistics: a computational study." En: *Behavior research methods* 39.3 (2007), págs. 510-26. ISSN: 1554-351X.

[8] John a Bullinaria y Joseph P Levy. "Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD." En: *Behavior research methods* 44.3 (2012), págs. 890-907. ISSN: 1554-3528. DOI: 10.3758/s13428-011-0183-8.

[9] John Caron. "Experiments with LSA scoring: Optimal rank and basis". En: *In Proceedings of SIAM Computational Information Retrieval Workshop* (2000), págs. 1-14.

[10] Kenneth Ward Church y Patrick Hanks. "Word association noms, Mutual Information, and lexicography". En: *Proceedings of the 27th Annual Conference of the Association for Computational Linguistics* 16.1 (1989), págs. 22-29. ISSN: 08912017. DOI: 10.3115/981623.981633.

[11] Ronan Collobert y Jason Weston. "A unified architecture for natural language processing". En: *Proceedings of the 25th international conference on Machine learning - ICML '08* 20.1 (2008), págs. 160-167. ISSN: 07224028. DOI: 10.1145/1390156.1390177.

[12] Ronan Collobert y col. “Natural Language Processing (almost) from Scratch”. En: *Journal of Machine Learning Research* 1 (2011), págs. 1-48. ISSN: 1532-4435. DOI: 10.1145/2347736.2347755. arXiv: 1103.0398.

[13] Scott Deerwester, Susan T. Dumais y Richard Harshman. “Indexing by latent semantic analysis”. En: *Journal of the American society for information science* 41.6 (1990), págs. 391-407. ISSN: 0002-8231. DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[14] Georgiana Dinu. “DISSECT-DIStributional SEmantics Composition Toolkit”. En: *Acl (2)* 2010 (2013), págs. 31-36.

[15] Susan T Dumais y col. “Using Latent Semantic Analysis to Improve Access to Textual Information”. En: *ACM Conference on Human Factors in Computing Systems, CHI '88* (1988), págs. 281-285. DOI: 10.1145/57167.57214.

[16] J L Elman. “Finding structure in time”. En: *Cognitive science* 14.2 (1990), págs. 179-211. ISSN: 03640213. DOI: 10.1207/s15516709cog1402{_}1.

[17] Lev Finkelstein y col. “Placing search in context: the concept revisited”. En: *ACM Transactions on Information Systems* 20.1 (2002), págs. 116-131. ISSN: 10468188. DOI: 10.1145/503104.503110.

[18] J. R. Firth. “Papers in Linguistics”. En: (1957).

[19] Edward Grefenstette. *AMA: Nando de Freitas*.

[20] Michael U Gutmann. “Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics”. En: *Journal of Machine Learning Research* 13 (2012), págs. 307-361. ISSN: 1532-4435.

[21] Zellig S. Harris. “Distributional Structure”. En: *Papers on Syntax*. Dordrecht: Springer Netherlands, 1954, págs. 3-22. DOI: 10.1007/978-94-009-8467-7{_}1.

[22] Felix Hill, Roi Reichart y Anna Korhonen. “SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation”. En: *Computational Linguistics* 41.4 (2015), págs. 665-695. ISSN: 04194217. DOI: 10.1162/COLI. arXiv: 1408.3456.

[23] Geoffrey E. Hinton. *Learning distributed representations of concepts*. 1986. DOI: 10.1109/69.917563.

[24] Thomas Hofmann. “Probabilistic Latent Semantic Analysis”. En: *Uncertainity in Artifitial Intelligence - UAI'99* (1999), pág. 8. ISSN: 15206882. DOI: 10.1.1.33.1187.

[25] S. Kaski. “Dimensionality reduction by random mapping: fast similarity\ncomputation for clustering”. En: *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)* 1.1 (1998), págs. 4-9. ISSN: 1098-7576. DOI: 10.1109/IJCNN.1998.682302.

[26] Thomas K Landauer y Susan T. Dumais. “A solution to Plato ' s problem : The Latent Semantic Analysis Theory of Acquisition , Induction , and Representation of Knowledge”. En: *Psychological Review* 104.2 (1997), págs. 211-240. ISSN: 0033-295X. DOI: 10.1037/0033-295X.104.2.211.

[27] O Levy, Y Goldberg e I Ramat-Gan. “Linguistic regularities in sparse and explicit word representations”. En: *CoNLL-2014* (2014), págs. 171-180.

[28] Omer Levy y Yoav Goldberg. "Neural Word Embedding as Implicit Matrix Factorization". En: (), págs. 1-9.

[29] Omer Levy, Yoav Goldberg e Ido Dagan. "Improving Distributional Similarity with Lessons Learned from Word Embeddings". En: *Transactions of the Association for Computational Linguistics* 3 (2015), págs. 211-225. ISSN: 2307-387X.

[30] Wei Li y Andrew McCallum. "Semi-Supervised Sequence Modeling with Syntactic Topic Models". En: *Proceedings of the 20th International conference on Artificial intelligence (AAAI 2005)* 2 (2005), págs. 813-818.

[31] Dekang Lin y Xiaoyun Wu. "Phrase clustering for discriminative learning". En: *Proceedings of the Joint Conference of the 47th Annual ...* August (2009), págs. 1030-1038. DOI: 10.3115/1690219.1690290.

[32] Kevin Lund y Curt Burgess. "Producing high-dimensional semantic spaces from lexical co-occurrence". En: *Behavior Research Methods, Instruments, & Computers* 28.2 (1996), págs. 203-208. ISSN: 0743-3808. DOI: 10.3758/BF03204766.

[33] Kevin Lund, Curt Burgess y Ruth Ann Atchley. "Semantic and Associative Priming in High-Dimensional Semantic Space". En: *Cognitive Science Proceedings, LEA* JANUARY 1995 (1995), págs. 660-665. ISSN: 18736009. DOI: 10.1016/j.jconhyd.2010.08.009.

[34] Risto Miikkulainen y Michael G Dyer. "Natural Language Processing with Modular Neural Networks and Distributed Lexicon". En: UCLA-AI-90-02 (1990).

[35] Tomáš Mikolov. "Language Models for Automatic Speech Recognition of Czech Lectures". En: *Proc. of STUDENT EEICT* 4 (2008).

[36] Tomas Mikolov, Wen-tau Yih y Geoffrey Zweig. "Linguistic regularities in continuous space word representations". En: *Proceedings of NAACL-HLT* June (2013), págs. 746-751.

[37] Tomas Mikolov y col. "Distributed Representations of Words and Phrases and their Compositionality". En: *Nips* (2013), págs. 1-9. ISSN: 10495258. DOI: 10.1162/jmlr.2003.3.4-5.951. arXiv: 1310.4546.

[38] Tomas Mikolov y col. "Efficient Estimation of Word Representations in Vector Space". En: *Proceedings of the International Conference on Learning Representations (ICLR 2013)* (2013), págs. 1-12. ISSN: 15324435. DOI: 10.1162/153244303322533223. arXiv: arXiv:1301.3781v3.

[39] Tomáš Mikolov y col. "Neural network based language models for highly inflective languages". En: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2009), págs. 4725-4728. ISSN: 15206149. DOI: 10.1109/ICASSP.2009.4960686.

[40] Tomáš Mikolov y col. "RNNLM — Recurrent Neural Network Language Modeling Toolkit". En: *Proceedings of ASRU 2011* (2011), págs. 1-4.

[41] Tomáš Mikolov y col. "Strategies for training large scale neural network language models". En: *2011 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2011, Proceedings.* 2011, págs. 196-201. ISBN: 9781467303675. DOI: 10.1109/ASRU.2011.6163930.

[42] Toma Mikolov y col. “Extensions of recurrent neural network language model”. En: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2011), págs. 5528-5531. ISSN: 15206149. DOI: 10.1109/ICASSP.2011.5947611.

[43] T Mikolov y col. “Recurrent Neural Network based Language Model”. En: *Interspeech* September (2010), págs. 1045-1048.

[44] a Mnih y Ge Hinton. “Three new graphical models for statistical language modelling.” En: *Proceedings of the 24th International Conference on Machine Learning (2007)* 62 (2007), págs. 641-648. DOI: 10.1145/1273496.1273577.

[45] Andriy Mnih y Geoffrey E. Hinton. “A Scalable Hierarchical Distributed Language Model.” En: *Advances in Neural Information Processing Systems* (2008), págs. 1-8.

[46] Andriy Mnih y Yee Whye Teh. “A Fast and Simple Algorithm for Training Neural Probabilistic Language Models”. En: *Proceedings of the 29th International Conference on Machine Learning (ICML'12)* (2012), págs. 1751-1758. arXiv: 1206.6426.

[47] Frederic Morin e Y Bengio. “Hierarchical probabilistic neural network language model”. En: *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (2005), págs. 246-252. DOI: 10.1109/JCDL.2003.1204852.

[48] Sebastian Padó, Ulrike Padó y Katrin Erk. “Flexible, corpus-based modelling of human plausibility judgements”. En: *Proceedings of EMNLP-CoNLL* 7.June (2007), págs. 400-409.

[49] Jeffrey Pennington, Richard Socher y Christopher Manning. “Glove: Global Vectors for Word Representation”. En: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, págs. 1532-1543. DOI: 10.3115/v1/D14-1162.

[50] Lizhen Qu y col. “Big Data Small Data , In Domain Out-of Domain , Known Word Unknown Word : The Impact of Word Representations on Sequence Labelling Tasks”. En: *CoNLL-2015* (2015), págs. 83-93. arXiv: 1504.05319.

[51] Kira Radinsky y col. “A word at a time: computing word relatedness using temporal semantic analysis”. En: *Proceedings of the 20th International World Wide Web Conference WWW'11* (2011), págs. 337-346. DOI: 10.1145/1963405.1963455.

[52] D. E. Rumelhart, G. E. Hinton y R. J. Williams. *Learning Internal Representations by Error Propagation*. 2013. DOI: 10.1016/B978-1-4832-1446-7.50035-2. arXiv: arXiv:1011.1669v3.

[53] Magnus Sahlgren. “Vector-based Semantic Analysis: Representing Word Meaning Based on Random Labels”. En: *ESSLI Workshop on Semantic Knowledge Acquisition and Categorization* (2002). DOI: 10.1.1.20.4588.

[54] Richard Socher, Jeffrey Pennington y Eh Huang. “Semi-supervised recursive autoencoders for predicting sentiment distributions”. En: *Conference on Empirical Methods in Natural Language Processing, EMNLP* i (2011), págs. 151-161. ISSN: 1937284115. DOI: 10.1.1.224.9432.

[55] Richard Socher, Alex Perelygin y Jy Wu. "Recursive deep models for semantic compositionality over a sentiment treebank". En: *Proceedings of the ...* (2013), págs. 1631-1642.

[56] Richard Socher y col. "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection". En: *Advances in Neural Information Processing Systems* (2011), págs. 801-809.

[57] Joseph Turian y col. "Word Representations: A Simple and General Method for Semi-supervised Learning". En: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* July (2010), págs. 384-394. DOI: 10.1.1.301.5840.

[58] Peter D Turney. "Mining the Web for synonyms: PMI-IR versus LSA on TOEFL". En: *Proceedings of the 12th European Conference on Machine Learning (ECML-2001), Freiburg, Germany* (2001), págs. 491-502. ISSN: 16113349. DOI: 10.1007/3-540-44795-4{_}42. arXiv: 0212033 [cs].

[59] Jaakko J Väyrynen y Timo Honkela. "Word Category Maps based on Emergent Features Created by ICA". En: ().

[60] Will Y Zou y col. "Bilingual Word Embeddings for Phrase-Based Machine Translation". En: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)* October (2013), págs. 1393-1398.