Capítulo 13

# Avances en representaciones de texto: espacios vectoriales semánticos y *word embeddings*[*]

Luis-Felipe Gutiérrez (https://orcid.org/0000-0001-8012-4836)
*luis.gutierrez@ucn.cl, Universidad Católica del Norte, Antofagasta, Chile.*

Brian Keith (https://orcid.org/0000-0001-5734-8962)
*brian.keith@ucn.cl, Universidad Católica del Norte, Antofagasta, Chile.*

## Introducción

El procesamiento de lenguaje natural (PLN) es un área de estudio que surge de la intersección entre inteligencia artificial y lingüística. La traducción palabra por palabra es una de las primeras soluciones propuestas. No obstante, los aspectos morfológicos más comunes de las palabras representan una de las causas del mal desempeño de las primeras soluciones a las tareas que involucran lenguaje natural (Nadkarni, Ohno-Machado & Chapman, 2011).

Si bien el PLN es un área activa en investigación, no existe consenso respecto de la definición del término. Se toma una de las propuestas aquí: PLN es un conjunto de técnicas computacionales que tienen el propósito de analizar y representar textos ocurridos naturalmente, en uno o más niveles de análisis lingüístico, con la finalidad de realizar procesamiento de lenguaje a nivel humano en un amplio rango de aplicaciones (Liddy, 2001).

Existen varios enfoques para realizar las tareas de PLN, tales como el simbólico basado en reglas (Lally & Fodor, 2011) y el simbólico basado en redes semánticas (Sowa, 2006). No obstante, en este capítulo se usa el enfoque basado en aprendizaje automático, que

[*] Investigación parcialmente financiada por la Comisión Nacional de Investigación Científica y Tecnología y el Ministerio de Educación del Gobierno De Chile. Proyecto REDI170607.

requiere la representación de los textos en forma matemática —en particular, representaciones vectoriales de los documentos y las palabras— (Le & Mikolov, 2014).

En la sección 1 del capítulo se aborda el concepto general de los espacios vectoriales semánticos; en la 2 se entrega una visión general de las técnicas de representación tradicionales en espacios vectoriales semánticos; la 3 comprende una revisión sistemática de la literatura de representaciones de documentos y palabras; en la 4 se detalla la discusión de resultados de dicha revisión, y en la 5 se resumen los elementos revisados y se entregan las conclusiones principales.

## 1. Espacios vectoriales semánticos

Los elementos que dan nombre a esta sección se usan para representar cada documento de una colección como un punto en el espacio, lo que equivale a un vector en un espacio vectorial. Así, los puntos que se encuentren cercanos entre ellos en el espacio comparten una semántica similar, mientras aquellos que se encuentren alejados tienden a ser semánticamente distintos. El término ‘semántica’ se relaciona con el significado de una palabra, frase u oración, o cualquier texto en lenguaje humano, además del estudio de tal significado (Mitchell & Lapata, 2008; Turney & Pantel, 2010; Jurafsky & Martin, 2014). Una vez se construye el espacio vectorial semántico, una tarea recurrente —dada una consulta, también llamada “pseudo-documento”— es comprobar con alguna medida de similitud cuáles documentos se le asemejan en términos semánticos.

Los vectores son estructuras comunes en el estudio de la inteligencia artificial y la ciencia cognitiva. Así, el aporte de los espacios vectoriales semánticos consiste en hacer uso de las frecuencias de ocurrencias de palabras en un cuerpo de texto para obtener información sobre su semántica (Turney & Pantel, 2010). Se detallan a continuación formas de representar la similitud entre documentos mediante distintos tipos de matrices (Turney & Pantel, 2010).

### *1.1. Matriz término-documento*

En el caso de contar con una amplia colección de documentos, y, por lo tanto, una gran cantidad de vectores de los mismos, es conveniente organizarlos en una matriz: en ella, las filas corresponden a términos (que usualmente son palabras); y las columnas, a los documentos.

En matemáticas discretas, una bolsa (también llamado multiconjunto, o *bag*, en inglés) es similar a un *set*, con la excepción de que las bolsas permiten duplicados y mantienen la propiedad de indiferencia en el orden de los elementos. No obstante, las bolsas normalmente no se representan usando la notación de conjuntos, sino que se utilizan vectores construidos con base en los elementos únicos y sus frecuencias de ocurrencia.

Un conjunto de bolsas se puede representar con una matriz X, en donde cada columna $x_j$ corresponde a una bolsa y cada fila $x_{i:}$ a un elemento único. Cada valor en la celda $x_{ij}$ es la frecuencia del i-ésimo elemento único en la j-ésima bolsa. En general, el valor de la mayoría de los elementos en X es cero, por lo que la matriz es poco poblada dado que la mayoría de los documentos contiene una pequeña fracción de la totalidad del vocabulario.

En una matriz termino-documento, un vector de documento representa el documento correspondiente como una bolsa de palabras (*bag of words*, en inglés). El método de representación anterior se basa en una hipótesis: la frecuencia de palabras en un documento tiende a indicar cuán relevante es para una consulta (pseudo-documento). Si los documentos y los pseudo-documentos tienen vectores similares en una matriz término-documento, existe una tendencia a tener significados similares (Salton, Wong & Yang, 1975); en palabras más simples, abordan temas parecidos.

Se debe notar que en este enfoque solo se entrega información sobre la frecuencia de las palabras del documento, pero no se tienen en cuenta el orden secuencial de las palabras ni elementos sintácticos o estructura de ningún tipo. No obstante, los vectores *bag of words*

suelen tener buen rendimiento en motores de búsqueda, por lo que logran capturar un aspecto importante de la semántica.

*1.2. Matriz palabra-contexto*

Bajo la misma lógica de la hipótesis de *bag of words*, si se examinan ahora los vectores fila de la matriz término-documento es posible medir la similitud entre palabras o términos. No obstante, si se desea obtener la similitud entre las primeras, la longitud óptima de texto para medirla no es necesariamente la que tienen los documentos completos; en lugar de ello se puede reducir el texto a una porción relativamente pequeña, denominada “contexto”. De esta manera, la matriz resultante es una de palabra-contexto, en la que el segundo término es un conjunto construido con base en palabras, oraciones, párrafos, capítulos o secuencias de caracteres y patrones. La matriz palabra-contexto se basa en la “hipótesis distributiva”, según la cual las palabras que ocurren en contextos similares tiendan a poseer similares significados. Lo anterior, en términos de la matriz palabra-contexto, equivale a decir que si algunas palabras tienen filas similares, son similares en términos semánticos (Harris, 1954; Firth, 1957; Deerwester *et al.*, 1990).

*1.3. Matriz par-patrón*

En una matriz par-patrón, los vectores fila corresponden a pares de palabras, tales como ‘pintor: lienzo’ y ‘carpintero: madera’, y los vectores columna corresponden a patrones en los cuales los pares de palabras coocurren. Lin y Pantel (2001) presentan las matrices par-patrón para medir la similitud semántica de patrones, utilizando los vectores columna de la matriz. La matriz par-patrón, se basa en la hipótesis distributiva extendida: esto es, los patrones que coocurren con pares de palabras similares tienden a tener significados similares. Si los patrones tienen vectores similares en una matriz par-patrón, potencialmente expresan relaciones semánticas parecidas (Lin & Pantel, 2001).

## 2. Representaciones tradicionales

El primer paso para determinar el espacio vectorial semántico consiste en escanear el corpus, contabilizar la ocurrencia de algún objeto (una palabra o par de palabras) en cierta situación (documento, contexto, o patrón) y almacenar el resultado en la entrada correspondiente en la matriz de ocurrencias. Se menciona la condición poco refinada que suele presentarse en las matrices de conteo de frecuencias, las cuales surgen tras realizar una etapa previa de preprocesamiento del corpus (Srividhya & Anitha, 2010). No obstante, existen otras propuestas de procesamientos posteriores para solucionar los inconvenientes que se presentan a continuación.

### *2.1. Term Frequency-Inverse Document Frequency (TF-IDF)*

TF-IDF es una medida en la que se asigna un peso a la frecuencia de los términos de una matriz de frecuencia, de tal manera que se asigne un peso mayor a aquellas palabras dadas que sean comunes a un documento o pequeño grupo de documentos en específico (Srividhya & Anitha, 2010). Así, las palabras que sean comunes a una gran porción de documentos tendrán un menor valor TF-IDF, pues su capacidad para diferenciar documentos entre sí es muy baja. Intuitivamente, este cálculo determina la relevancia de una palabra para un documento dado.

### *2.2. Pointwise Mutual Information (PMI)*

PMI permite comparar de modo intuitivo la probabilidad de observar una palabra (punto) *i* y una palabra *j* juntas (probabilidad conjunta de observar *i* y *j*), con la probabilidad de observar *i* y *j* de manera independiente. Así, si existe una asociación genuina entre *i* y *j*, la probabilidad conjunta $p_{ij}$ será mucho mayor que el valor $p_i p_j$. La interpretación matemática de esta condición se observa en la formalización del método PMI (Deerwester *et al.*, 1990).

### *2.3. Singular Value Decomposition (SVD)*

Deerwester (1990) propone un método para mejorar la medida de similitud en una matriz término-documento X basado en álgebra lineal. La operación en la que se basa se denomina SVD truncado. El autor menciona que esta técnica se puede aplicar para determinar similitud entre palabras —*Latent Semantic Analysis* - LSA— y entre documentos —*Latent Semantic Indexing* - LSI— (Turney & Pantel, 2010).

En este caso, la descomposición en valores singulares se realiza sobre la matriz término-documento X, factorizándola en tres matrices diferentes de la manera $X = U\Sigma V^T$, en donde U y V son ortonormales en cuanto a sus columnas, y Σ es una matriz diagonal de valores singulares (Golub & Van Loan, 2012). SVD se puede entender desde distintas perspectivas:

- Significado latente: al limitar el número de dimensiones latentes, se fuerza una mayor correspondencia entre palabras y contextos (Deerwester *et al.*, 1990).
- Reducción de ruido: Rapp (2003) presenta el método SVD como un esquema de este tipo.
- Coocurrencia de alto nivel: Landauer y Dumais (1997) describen un método que usa SVD para descubrir coocurrencias de alto nivel (es decir, cuando dos palabras se encuentran en contextos similares).
- Reducción de dispersión: en general, la matriz X es poco poblada; pero al aplicar SVD truncado, la matriz resultante es densa.

## 3. Revisión sistemática de la literatura

Existen propuestas para representar palabras mediante vectores densos que se derivan de diversos métodos de entrenamiento, inspirados en el modelado de lenguajes mediante redes neuronales. Este tipo de representaciones se denomina *neural embeddings* o *word embeddings* (Levy & Goldberg, 2014). Estas demuestran ser un mecanismo con el cual se facilita la tarea de computar similitudes entre palabras, mediante cálculos eficientes con

operaciones de matrices de baja dimensionalidad. Además, son eficientes en cuanto a entrenamiento y altamente escalables, tanto para corpus de gran tamaño (miles de millones de palabras) como para vocabularios y contextos de similares proporciones (Levy & Goldberg, 2014).

Las representaciones densas de palabras en espacios vectoriales semánticos ocupan un rol importante para tareas básicas del PLN (Collobert & Weston, 2008; Zou *et al.*, 2013), al igual que para tareas más complejas —por ejemplo, el análisis de sentimientos— (Severyn & Moschitti, 2015; Tang *et al.*, 2014).

La revisión sistemática de la literatura se realizó según el método de Kitchenham (2004), el cual involucra las etapas de planificación, ejecución y reporte de la investigación. Los resultados de la revisión se muestran en la segunda columna de la tabla 1. Luego del análisis sistemático de los artículos recuperados, la cantidad de artículos seleccionados se muestra en la tercera columna de la misma tabla.

*Tabla 1.* Cantidad de artículos encontrados y seleccionados por fuente

| Fuente de artículos | Cantidad de artículos encontrados | Cantidad de artículos seleccionados |
|---|---|---|
| Web of Science | 11 | 4 |
| SCOPUS | 125 | 34 |
| ACM Digital Library | 14 | 7 |
| Total | 150 | 45 |

Fuente: elaboración propia.

## 4. Discusión de resultados

Los 45 artículos seleccionados en esta revisión sistemática posicionan las *word embeddings* como una técnica ubicua en el PLN en general. A continuación se mencionan los artículos relevantes y el uso que en ellos se da al concepto de *word embeddings*, comenzando con las dos representaciones de palabras encontradas en casi la totalidad de los artículos, usadas directamente para la realización de las tareas o como línea base para la evaluación de nuevos modelos de representación.

En primer lugar se deben destacar los modelos *word2vec,* nombre con el que se conoce a dos modelos de lenguaje basados en redes neuronales propuestos por Mikolov *et al.* (2013) para generar representaciones vectoriales densas de palabras. De modo específico, se proponen dos arquitecturas de modelos: *Continuous Bag of Words Model* (CBOW) y *Continuous Skip Gram Model* (SG). Por un lado, CBOW tiene como objetivo predecir la ocurrencia de una palabra dadas otras palabras que constituyen su contexto. Se entiende por contexto de una palabra $w_i$, a la vecindad compuesta por las *k* palabras a la izquierda de $w_i$ y las *k* a la derecha de $w_i$. Por otro lado, SG se ocupa de predecir un contexto, dada la palabra $w_i$. En este modelo, *k* es un hiperparámetro conocido como tamaño de la ventana de contexto local. En ambos casos, los modelos entregan representaciones vectoriales densas para las palabras que demuestran conservar las características semánticas, a pesar de una drástica reducción de la dimensionalidad y entrenamiento en una red neuronal poco profunda, lo que, además, agiliza este proceso.

La segunda representación vectorial recurrente en los artículos se denomina *Global Vectors for Word Representation* – GloVe (Pennington, Socher & Manning, 2014). A diferencia de *word2vec*, el cual es un modelo predictivo, GloVe es más cercano a un modelo que permite reducir la dimensionalidad de una matriz de coocurrencia del tipo palabra-palabra, que se genera con una ventana de contexto local de dimensión fija. GloVe recibe su nombre debido a que las estadísticas de todo el corpus (a nivel global) se capturan directamente del modelo. Además, es competitivo y reporta mejores resultados que métodos como *word2vec* en tareas como la analogía y similitud de palabras, o reconocimiento de entidades.

Uno de los principales usos reportados para las *word embbedings* es la evaluación de similitud semántica entre palabras de distintos idiomas, lo cual, en esencia, se remonta a

las primeras aplicaciones del PLN. En este sentido, Vulić & Moens (2015) proponen un modelo para aprender de manera conjunta *word embeddings* bilingües, con base solo en datos comparables constituidos con documentos alineados que se encuentran en dos idiomas distintos. Este modelo, denominado *Bilingual Word Embeddings Skip Gram* (BWESG), induce un espacio vectorial multilingüe para embeber representaciones de palabras, consultas e incluso documentos completos. En esta misma línea, Glavaš *et al*. (2017) proponen otro método para medir similitud semántica textual entre documentos escritos en distintos idiomas: el método ligero en cuanto al uso de recursos, consiste en trasladar linealmente representaciones de palabras de un espacio vectorial en un idioma de origen al espacio vectorial del idioma de destino. Las *word embeddings* usadas en dicho trabajo se generan mediante GloVe y CBOW.

Existen también propuestas de *word embeddings* para lenguas que cuentan con alfabetos particulares, tales como el árabe. Soliman, Eissa y El-Beltagy (2017) proponen un conjunto preentrenado de modelos de representaciones de palabras en el idioma árabe, con el fin de otorgar a la comunidad *word embeddings* generadas a partir de dominios como tuits, páginas de Internet y artículos de Wikipedia en árabe. También se plantean soluciones a la desambiguación de palabras, tarea recurrente en el PLN, mediante el uso de *word embeddings* en árabe. De modo específico, Laatar, Aloulou y Bilguith (2017) proponen esta solución con el fin de elaborar un diccionario que muestre la evolución del significado y uso de palabras árabes, que a su vez puede ayudar a salvaguardar la herencia cultural árabe. Se debe destacar que estos artículos se basan en *word embeddings* generadas con *word2vec*.

Otro uso común de las *word embeddings* consiste en su incorporación en sistemas de recomendación, es decir, herramientas de *software* que proveen sugerencias y recomendaciones a un usuario particular (Ricci, Rokach & Shapira, 2015). En este sentido, Musto *et al.* (2016) presentan resultados preliminares para la adopción de *word embeddings,* en los

que tanto objetos como perfiles de usuarios se embeben en un espacio vectorial para utilizarlos en un sistema de recomendación basado en contenidos.

Boratto *et al.* (2016), por su parte, plantean que el uso de *word embeddings* en sistemas de recomendación basados en contenidos es menos efectivo que otras estrategias colaborativas (p. ej. descomposición de valores singulares). Proponen entonces la definición de un espacio vectorial en el cual la similitud entre un objeto que no evalúa el usuario y aquellos que sí lo son se mide en términos de independencia lineal, con lo que se obtienen mejores resultados que SVD, por ejemplo.

Greenstein-Messica, Rokach y Friedman (2017) plantean la conversión a palabras de la secuencia de objetos que busca un usuario para proyectarlas en un espacio vectorial, de tal manera que se pueda detectar similitud y analogías entre objetos. Las *word embeddings* se generan según los modelos *word2vec* y GloVe.

También se reporta el uso de *word embeddings* en conjunto con otras técnicas de aprendizaje automático o recursos lingüísticos. Según Alsuhaibani *et al.* (2018), los métodos que permiten generar representaciones vectoriales de palabras con base solo en información distribuida en un corpus desaprovechan la estructura relacional semántica que se presentan entre palabras en contextos coocurrentes. Dichas estructuras se detallan en bases de conocimiento elaboradas manualmente, tales como ontologías y léxicos semánticos, en los que el significado de las palabras se define mediante las diversas relaciones que existen entre ellas. Por esto se combina el corpus con las bases de conocimiento para generar *word embeddings* que, al usarse, presenten una mejora del rendimiento en las tareas de medición de similitud y analogía de palabras, y hagan posible obtener resultados que avalen la hipótesis.

A su turno, Liu (2017) propone que además de generar representaciones vectoriales de palabras teniendo como fuente el corpus, también se deben considerar elementos internos

de cada palabra, como los morfemas. En razón de ello, propone dos modelos para generar *word embeddings*: *Morpheme on Original view and Morpheme on Context view* (MOMC) y *Morpheme on Context view* (MC), que poseen mayor rendimiento para detectar similitud de palabras que los modelos de la línea base —entre los que se encuentra CBOW—. Gallo, Nawaz y Calefati (2017) presentan un método en el que las *word embeddings* generadas con *word2vec* se codifican en imágenes, para luego hacer uso de redes neuronales convolucionales (CNN) y realizar clasificación de texto sobre las imágenes. Con el método se reportan mejores resultados de clasificación al compararlos con la línea base (*doc2vec* con SVM).

Wild y Stahl (2007) presentan una implementación de *Latent Semantic Analysis* y sus resultados al generar representaciones de palabras en espacios vectoriales. A diferencia de *word2vec* y GloVe, este método se basa en la factorización de matrices debido al uso de la descomposición de valores singulares.

Uno de los principales inconvenientes que presentan las *word embeddings* consiste en la disminución de dimensionalidad del problema a costa de la interpretabilidad de los valores reales que implican las representaciones vectoriales, lo que comúnmente se conoce como modelos opacos. Ante esto, en la literatura se proponen soluciones para hacer las representaciones interpretables. Por ejemplo, Liu *et al.* (2018) proponen técnicas para visualizar analogías semánticas y sintácticas útiles en diversos dominios, usando como representaciones base las generadas con *word2vec* y GloVe. Andrews (2016) señala que a pesar de ostentar dimensiones reducidas, en las representaciones aprendidas con los modelos de *word embeddings* se hace uso de una cantidad no menor de almacenamiento, por lo que se propone el uso del algoritmo de Lloyd para comprimir las representaciones densas en un factor de 10 sin penalizar de mayor manera el rendimiento. Además, se presenta un método de factorización eficiente de cómputo en GPU para obtener representaciones con

mayor interpretabilidad, al tener cada dimensión codificada con un valor no negativo. Al evaluar las tareas de similitud y analogía de palabras con las representaciones comprimidas, se demuestra que las aspiraciones del trabajo son alcanzables.

En cuanto a la eficiencia computacional, hay autores que abordan la minimización del costo de entrenamiento. En particular, Joulin *et al.* (2016) presentan un método para generar *word embeddings* de varios órdenes de magnitud más rápido que los modelos basados en aprendizaje profundo. El método es similar a CBOW: se reemplaza la palabra del medio del contexto con una etiqueta de clasificación, con lo que se obtiene un rendimiento a la par con modelos que tardan más tiempo en entrenamiento.

Finalmente, Moody (2016) presenta un método en el que se mezcla la arquitectura *Skip Gram* de *word2vec* con el modelado de tópicos en documentos, en el que se usa la técnica *Latent Dirichlet Allocation*. El modelo se denomina *lda2Vec* y permite generar representaciones de palabras y documentos en un mismo espacio vectorial.

## 5. Conclusiones

En este capítulo se realizó una visión general de los modelos de representación de documentos y palabras en el contexto del procesamiento de lenguaje natural. Las representaciones vectoriales densas de palabras se adoptan ampliamente con resultados satisfactorios en tareas de procesamiento de lenguaje natural en general y, además, se aplican en otros dominios con buenos resultados.

En este contexto, se reporta la hegemonía de las *word embeddings* basadas en modelos neuronales sobre las generadas con factorización de matrices. En la literatura se proponen numerosas alternativas que son variantes de un grupo de modelos neuronales *word2vec*. En este contexto, se condujeron estudios sobre el impacto que genera la inclusión de otras técnicas junto con los *word embeddings* para realizar tareas de procesamiento de lenguaje natural, reportando buenos resultados. Dentro de la revisión sistemática, se constató la gran

cantidad de artículos que utilizan los modelos *word2vec,* de tal forma que se podrían considerar como el estándar *de facto* en estos momentos.

A pesar del buen rendimiento de los *word embeddings*, se identifican algunos inconvenientes y sus respectivas propuestas de solución —como la falta de interpretabilidad de los valores reales que componen los vectores embebidos—. Por otro lado, los resultados reportados en el uso de los *word embeddings* dan importancia a esta técnica en PLN, considerando el planteamiento matemático de los modelos, la aparición de mejoras en diferentes aspectos de los modelos iniciales (p. ej. tiempo de entrenamiento) y su incorporación con otras técnicas utilizadas en tareas de PLN.

## Referencias

Alsuhaibani, M., Bollegala, D., Maehara, T. & Kawarabayashi, K. I. (2018). Jointly learning word embeddings using a corpus and a knowledge base. *PloS one*, *13*(3), e0193094.

Andrews, M. (2016, October). Compressing word embeddings. En *International Conference on Neural Information Processing* (pp. 413-422). Springer, Cham.

Boratto, L., Carta, S., Fenu, G. & Saia, R. (2016). Representing Items as Word-Embedding Vectors and Generating Recommendations by Measuring their Linear Independence. En *RecSys Posters*.

Collobert, R. & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. En *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*. Blackwell, Oxford.

Gallo, I., Nawaz, S. & Calefati, A. (2017, November). Semantic Text Encoding for Text Classification Using Convolutional Neural Networks. En *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on* (Vol. 5, pp. 16-21). IEEE.

Glavaš, G., Franco-Salvador, M., Ponzetto, S. P. & Rosso, P. (2017). A resource-light method for cross-lingual semantic textual similarity. *Knowledge-Based Systems*.

Golub, G. H. & Van Loan, C. F. (2012). *Matrix computations* (Vol. 3). JHU Press.

Greenstein-Messica, A., Rokach, L. & Friedman, M. (2017, March). Session-based recommendations using item embedding. En *Proceedings of the 22nd International Conference on Intelligent User Interfaces* (pp. 629-633). ACM.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146-162.

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jurafsky, D. & Martin, J. H. (2014). *Speech and language processing* (Vol. 3). London: Pearson.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*(2004), 1-26.

Laatar, R., Aloulou, C. & Bilguith, L. H. (2017). Word sense disambiguation of Arabic language with Word Embeddings as part of the Creation of a Historical Dictionary.

Lally, A. & Fodor, P. (2011). Natural language processing with prolog in the IBM Watson system. *The Association for Logic Programming (ALP) Newsletter*.

Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Le, Q. & Mikolov, T. (2014, January). Distributed representations of sentences and documents. En *International Conference on Machine Learning* (pp. 1188-1196).

Levy, O. & Goldberg, Y. (2014). Dependency-based word embeddings. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Vol. 2, pp. 302-308).

Liddy, E. D. (2001). *Natural language processing*. Surface.

Lin, D. & Pantel, P. (2001, August). DIRT – discovery of inference rules from text. En *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 323-328). ACM.

Liu, J. (2017). Morpheme-Enhanced Spectral Word Embedding. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE,* 551-556.

Liu, S., Bremer, P. T., Thiagarajan, J. J., Srikumar, V., Wang, B., Livnat, Y. & Pascucci, V. (2018). Visual Exploration of Semantic Relationships in Neural Word Embeddings. *IEEE transactions on visualization and computer graphics*, *24*(1), 553-562.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mitchell, J. & Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, 236-244.

Moody, C. E. (2016). Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

Musto, C., Semeraro, G., de Gemmis, M. & Lops, P. (2016, March). Learning word embeddings from wikipedia for content-based recommender systems. En *European Conference on Information Retrieval* (pp. 729-734). Springer.

Nadkarni, P. M., Ohno-Machado, L. & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544-551.

Pantel, P. & Lin, D. (2002, July). Discovering word senses from text. En *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 613-619). ACM.

Pennington, J., Socher, R. & Manning, C. (2014). Glove: Global vectors for word representation. En *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. En *Proceedings of the ninth machine translation summit* (pp. 315-322).

Ricci, F., Rokach, L. & Shapira, B. (2015). Recommender systems: introduction and challenges. En *Recommender systems handbook* (pp. 1-34). Springer, Boston, MA.

Salton, G., Wong, A. & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

Severyn, A. & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. En *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 959-962). ACM.

Soliman, A. B., Eissa, K. & El-Beltagy, S. R. (2017). AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. *Procedia Computer Science*, *117*, 256-265.

Sowa, J. F. (2006). Semantic networks. *Encyclopedia of Cognitive Science*.

Srividhya, V. & Anitha, R. (2010). Evaluating preprocessing techniques in text categorization. *International journal of computer science and application*, *47*(11), 49-51.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T. & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 1555-1565).

Turney, P. D. (2001, September). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. En *European Conference on Machine Learning* (pp. 491-502). Springer, Berlin, Heidelberg.

Turney, P. D. & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*, 141-188.

Vulić, I. & Moens, M. F. (2015, August). Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. En *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 363-372). ACM.

Wild, F. & Stahl, C. (2007). Investigating unstructured texts with latent semantic analysis. En *Advances in Data Analysis* (pp. 383-390). Berlín: Springer.

Zou, W. Y., Socher, R., Cer, D. & Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* 1393-1398.