Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

Tipo de artículo: Artículo de revisión
Temática: Reconocimiento de patrones
Recibido: 19/11/2015 | Aceptado: 18/03/2016

Representación textual en espacios vectoriales semánticos

Textual representation in semantic vector space

Carmen Torres López [1*], Leticia Arco García [2]

[1] Desoft-Holguín. carmentorreslopez87@gmail.com
[2] Universidad Central "Marta Abreu" de Las Villas. leticiaa@uclv.edu.cu

[*] Autor para correspondencia: leticiaa@uclv.edu.cu

Resumen

El modelo espacio vectorial representa documentos textuales a través de vectores de términos, pero no permite representar relaciones semánticas entre las palabras. Los espacios vectoriales semánticos se basan en la idea que el significado de una palabra puede ser aprendido de un entorno lingüístico y poseen dos enfoques, la semántica distribucional y la semántica composicional. El primer enfoque analiza el significado de palabras individuales y el segundo enfoque el significado de frases, oraciones y párrafos. La presente revisión expone los principales modelos de estos dos enfoques, así como las herramientas computacionales que los implementan. Como resultado de este estudio se concluye que se hace necesario la incorporación de representaciones semánticas en las distintas herramientas que realizan análisis textual, fundamentalmente han dado mejores resultados aquellas representaciones que hacen predicción de contextos para el caso de modelos distribucionales y las que incorporan modelos basados en redes neuronales para los modelos composicionales.

Palabras clave: minería de textos, espacio vectorial semántico, semántica distribucional, semántica composicional

Abstract

The vector space model represents textual documents via vectors of terms, but it cannot represent semantic relationships between words. Semantic vector spaces are based on the idea that the meaning of a word can be learned from a linguistic environment and have two approaches, the distributional semantics and compositional semantics. The first approach analyzes the meaning of individual words and the second approach the meaning of phrases, sentences and paragraphs. This review presents the main models of these approaches and the computational tools that implement them. This study bring to a conclusion that the incorporation of semantic representations in the different tools that perform textual analysis is necessary, essentially researchers have obtained best representations that make prediction

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

of contexts in the case of distributional models and the ones that incorporate models based on neural networks for compositional models.

Keywords: *text mining, semantic vector space, distributional semantics, compositional semantics*

Introducción

En la actualidad es inmensa la cantidad de datos generados por los usuarios en los medios computacionales. El desarrollo constante de nuevas tecnologías y programas informáticos es una de las principales causas del problema de cómo manejar tales cantidades de datos y cómo obtener de forma eficiente el conocimiento que buscan los usuarios en distintos dominios. El formato más común de almacenamiento es el texto y son varios los modelos propuestos para representarlo. Uno de ellos es el modelo espacio vectorial y son varios los investigadores que hacen énfasis en el estudio de cómo incluir elementos semánticos en este modelo. Conocer el significado de los textos a través de algoritmos computacionales es un reto que conlleva grandes beneficios en diversos contextos, por ejemplo, para la tarea del análisis de sentimiento, la traducción automática de idiomas, la clasificación de documentos de acuerdo a su contenido, para sistemas de recomendación y detección de tópicos, entre otras. Todas estas tareas tienen aplicación en dominios científicos, médicos, de producción, etc.

La representación semántica de los textos se ha generalizado en tres grupos, las redes semánticas, los modelos basados en rasgos y los espacios semánticos (Mitchell & Lapata, 2010). El primer grupo representa los conceptos como nodos de un grafo y las aristas son las relaciones semánticas entre los conceptos; el significado de una palabra es expresado por la cantidad y tipo de conexiones con otras palabras. El segundo grupo sigue la idea que el significado de las palabras puede ser descrito por listas de rasgos, en algunos casos se crean manualmente y en otros casos se obtienen atributos facilitados por hablantes nativos; esto permite una representación de cada palabra a través de una distribución de valores numéricos sobre un conjunto de rasgos. El tercer grupo estudia las representaciones semánticas basado en la idea que el significado de una palabra puede ser aprendido de un entorno lingüístico. Los modelos de espacio semántico capturan el significado cuantitativamente en términos de estadística de coocurrencia simple. Otra variante semántica son los modelos de tópicos probabilísticos que se basan en que las palabras observadas en un corpus poseen una estructura latente enlazada a tópicos (Mitchell & Lapata, 2010).

En la literatura de los espacios semánticos existen dos tendencias importantes, la semántica distribucional y la semántica composicional. Para la primera se han propuesto varios modelos de representación basados en matrices con el objetivo de modelar el significado de las palabras utilizando el modelo espacio vectorial y para la segunda se han propuesto

otros modelos más amplios enfocados a modelar el significado no solo de las palabras, sino de frases y oraciones. Recientemente han surgido herramientas en varios lenguajes de programación que facilitan la modelación de algoritmos que siguen estos modelos.

A pesar de que existen varios trabajos que abordan técnicas para la representación textual incorporando elementos semánticos, así como algunas herramientas, marcos de trabajo y bibliotecas que incorporan algunas de las formas de representación publicadas, aún es insuficiente la publicación de trabajos que consoliden las técnicas y herramientas existentes así como las características principales de cada una de ellas, sus ventajas y desventajas, de forma tal que sea fácil para un investigador conocer qué variantes existen y cómo aplicarlas. De ahí que el presente artículo tiene como objetivo mostrar el resultado de una revisión de la literatura relacionada a la construcción de modelos, algoritmos y utilización de herramientas computacionales para la representación de textos en espacios vectoriales semánticos. De esta forma, los investigadores en el área de la minería de textos, procesamiento del lenguaje natural y minería de opinión, tendrán los elementos suficientes para identificar aquellas formas de representación que mejor se adapten a la problemática a tratar, sobre todo considerando las bondades de incluir elementos semánticos en la representación para potenciar la calidad de los resultados de algoritmos de procesamiento textual que posteriormente se apliquen.

1. Espacios vectoriales semánticos

Los textos son datos no estructurados de gran dimensionalidad. Varios han sido los modelos computacionales propuestos para la representación textual, ejemplos de estos modelos son: el modelo booleano (Baeza-Yates & Ribeiro-Neto, 1998), el modelo espacio vectorial (Salton et al., 1975), el análisis semántico latente (Deerwester, 1988) y los grafos (Biggs, N.; Lloyd, E. Wilson, 1986), entre otros.

El modelo espacio vectorial (Vector Space Model; VSM) representa documentos textuales a través de vectores de términos. Fue presentado por Gerard Salton y sus colegas en 1975 (Salton et al., 1975) y desde entonces es uno de los modelos de representación más usados en tareas de recuperación de información y procesamiento del lenguaje natural (Natural Language Processing; NLP). Una interpretación de este modelo es: "*En VSM cada documento se identifica como un vector de rasgos en un espacio en el cual cada dimensión corresponde a términos indexados distintos (palabras). Un vector documento dado, en cada componente tiene un valor numérico para indicar su importancia*" (Arco, 2008). VSM utiliza el enfoque lingüístico léxico, el cual se refiere al análisis concerniente a palabras individuales; y se basa en dos elementos fundamentales: un esquema de pesos y una medida de similitud. VSM se fundamenta en una comparación estricta de los términos, por lo que opera en el plano estadístico de los documentos, es

decir, considera los documentos como bolsas de palabras. Estas características constituyen una limitante para el modelo, debido a que no permite representar relaciones semánticas entre las palabras.

El nivel semántico es un nivel lingüístico que analiza el significado de una palabra o frase. Por ejemplo, al comprender frases permite conocer el significado de varias palabras en su conjunto. Existe una estrecha relación entre el enfoque léxico y el semántico, de ahí que en (Manning, 1999) (Jurafsky & Martin, 2007) se hace referencia que el enfoque semántico se divide en dos partes: el estudio del significado de palabras individuales (semántica léxica) y de cómo los significados de palabras individuales son combinados en el significado de oraciones o incluso unidades más grandes (semántica composicional).

Los primeros pasos para incorporar la semántica en el procesamiento del lenguaje natural están muy relacionados al estudio del espacio vectorial y al álgebra lineal. Varios aspectos semánticos fundamentalmente desde el punto de vista léxico están relacionados geométricamente por una noción de distancia. Por ejemplo, el significado de la palabra "gato" está más cerca al significado de la palabra "perro" que al significado de la palabra "carro" (Clark, 2014). Los significados de las palabras pueden ser representados usando vectores, como parte de un espacio semántico de alta dimensión. La estructura detallada de este espacio se provee al considerar los contextos[1] en los cuales las palabras ocurren en un gran corpus textual. Las palabras son fácilmente comparadas mediante similitudes en el espacio vectorial, usando cualquiera de las medidas de distancia del álgebra lineal, una de las más comunes es la medida coseno, la cual calcula el coseno del ángulo entre dos vectores. Estas ideas se resumen en la llamada metáfora geométrica del significado: "*Los significados son ubicaciones en un espacio semántico, y la similitud semántica representa la proximidad entre las ubicaciones*" (Sahlgren, 2006). De esta forma, la proximidad espacial entre palabras indica cuán similares son sus significados. Los llamados modelos espacio vectorial de significado son también conocidos por modelos de espacio-palabra (Sahlgren, 2006). Actualmente existen varias implementaciones de estos modelos, a continuación se expondrán las principales tendencias y las herramientas computacionales en las que se encuentran.

1.1 Semántica distribucional

En 1957 el lingüista John R. Firth sentó las bases de la teoría distribucional moderna con la idea: "*usted conocerá una palabra por la compañía que posee*" (Firth, 1957). La semántica distribucional se basa en obtener patrones estadísticos

[1] El escenario de una palabra, frase, etc. entre las palabras, frases que lo rodean. A menudo es usado para ayudar a explicar el significado de una palabra, frase, etc. (Sahlgren 2006).

de las palabras (como la coocurrencia de palabras), a partir de los cuales se descubren las diferencias o similitudes entre ellas. A continuación se describirán los tipos de modelos distribucionales semánticos y sus principales aspectos.

1.1.1 Modelos distribucionales semánticos basados en vectores de conteo

Los modelos distribucionales basados en vectores de conteo siguen cuatro etapas fundamentales para obtener las coocurrencias de términos o palabras en los documentos de un corpus: realizar una representación del texto para extraer la cantidad de coocurrencias; utilizar un esquema de pesos para estimar dichas cantidades; reducir la dimensionalidad de las representaciones y comparar las unidades textuales a través de medidas de similitud. Estas etapas se describen a continuación (Grefenstette, Moritz, et al., 2014).

1.1.1.1 Extracción de cantidades de coocurrencia

Para representar la frecuencia de aparición de unidades lingüísticas en un texto se construye una matriz. En la literatura se encuentran de forma general tres tipos de matrices, las cuales analizan similitudes de documentos (matrices término-documento), de palabras (matrices palabra-contexto) y de relaciones (matrices par-patrón) (Turney, 2010).

Las matrices término-documento presentan filas que corresponden a términos y las columnas a documentos, en este caso un vector de documentos se representa como una bolsa de palabras. Esta es una de las formas más comunes de modelar documentos, en la cual se cuenta la cantidad de ocurrencias de cada término pero se ignora el orden en que aparecen, es decir, que la estructura lingüística del texto se desconoce. Por lo general, la mayor parte de los elementos de esta matriz son 0, por lo que es una matriz dispersa, debido a que la mayoría de los documentos usarán solo una fracción de todo el vocabulario. En este tipo de matrices los vectores columna similares indican documentos similares. Las matrices término-documento son muy usadas en el área de recuperación de información, donde la hipótesis de bolsa de palabras captura en cierta medida el tema que trata el documento.

Las matrices palabra-contexto presentan filas que corresponden a términos y las columnas a contextos. El contexto está dado por palabras, frases, oraciones, párrafos, capítulos, documentos u otras posibilidades como secuencias de caracteres o patrones. Esta representación se basa en la hipótesis de distribución en lingüística que plantea que las palabras que ocurren en contextos similares tienden a tener significados similares. En este tipo de matrices los vectores fila similares sugieren palabras con significados similares. En áreas como la recuperación de información, el contexto se observa como todo el documento, sin embargo puede ser reducido a una oración o incluso a algunas palabras cercanas a la palabra de la cual se quieren obtener palabras similares (palabra objetivo). De esta forma se obtienen las llamadas matrices término-término o palabra-palabra, en las cuales se consideran palabras únicas como contexto y se cuenta la cantidad de veces que una palabra contexto ocurre en el contexto de una palabra objetivo (Clark, 2014). Para el análisis

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

del contexto se han estudiado dos tipos de relaciones: palabras relacionadas sintagmáticamente[2] (palabras que ocurren en el mismo documento) y palabras relacionadas paradigmáticamente (palabras que ocurren cerca una de otra, es conocido por coocurrencia léxica). Se ha afirmado que el segundo enfoque revela mayor información y por tanto provee mejor base estadística (Sahlgren, 2006).

An automobile is a wheeled motor vehicle used for transporting passengers .
A car is a form of transport, usually with four wheels and the capacity to carry around five passengers .
Transport for the London games is limited , with spectators strongly advised to avoid the use of cars .
The London 2012 soccer tournament began yesterday , with plenty of goals in the opening matches .
Giggs scored the first goal of the football tournament at Wembley , North London .
Bellamy was largely a passenger in the football match , playing no part in either goal .

Figura 1. Texto de ejemplo. Fuente: (Clark 2014)

Term vocab: $\langle wheel,\ transport,\ passenger,\ tournament,\ London,\ goal,\ match \rangle$

$$\begin{array}{l} \\ automobile \\ car \\ soccer \\ football \end{array} \begin{array}{c} \begin{array}{ccccccc} wheel & transport & passenger & tournament & London & goal & match \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 & 1 \end{pmatrix} \end{array}$$

$automobile\ .\ car = 4$
$automobile\ .\ soccer = 0$
$automobile\ .\ football = 1$
$car\ .\ soccer = 1$
$car\ .\ football = 2$
$soccer\ .\ football = 5$

Figura 2. Ejemplo de matriz término-término. Fuente: (Clark 2014)

[2] Un sintagma es una combinación de entidades lingüísticas ordenadas, por ejemplo, las oraciones son sintagmas de palabras y los párrafos son sintagmas de oraciones. Las relaciones de paradigmas se refieren a sustituciones y relacionan entidades que no coocurren en el texto, un paradigma es un conjunto de entidades que se pueden sustituir (Sahlgren 2006).

En la Figura 2 se muestra un ejemplo de una matriz término-término, para el texto correspondiente a la **¡Error! No se encuentra el origen de la referencia.**, el contexto está formado por términos extraídos de un conjunto de oraciones[3]. Las palabras objetivos (para las que los vectores contexto son calculados) no tienen que ser parte del vocabulario de términos que provee el contexto. Para determinar palabras similares puede usarse la medida coseno. En el ejemplo, "football" es similar en significado a "soccer" debido a que el vector contexto (fila) correspondiente a "football", superpone el vector correspondiente a "soccer", es decir que varias de las palabras que rodean a "football" son las mismas que rodean a "soccer"; en el ejemplo estas dos palabras coinciden con una frecuencia igual a 5 para los contextos especificados (Clark 2014). Los coeficientes de la matriz representan la frecuencia de los términos en las oraciones.

Las matrices par-patrón presentan filas que corresponden a pares de palabras, ejemplo carpintero:madera, y los vectores columna corresponden a los patrones en los que los pares coocurren, por ejemplo, "X corta Y", "Y es cortado por X" coocurre con el par X:Y (la cantidad de patrones no tiene que coincidir con la cantidad de documentos a los que pertenecen los pares) (Turney, 2010). En este caso el objetivo es medir la similitud semántica de patrones (de los vectores columna). Esta representación se basa en la hipótesis de distribución extendida, donde los patrones que coocurren con pares similares tienden a tener significados similares, y por tanto puede ser usada, por ejemplo, para inferir que una oración es una paráfrasis de otra. Otros estudios introducen pares de palabras con vectores filas similares que tienden a tener relaciones semánticas similares, es decir, similitud a través de los vectores fila, por ejemplo albañil:piedra, carpintero:madera y alfarero:arcilla comparten la relación semántica artesano:material y los patrones son "X usa Y para" y "X transforma Y en". Otra hipótesis propuesta es la de relaciones latentes, donde pares de palabras que coocurren en patrones similares tienden a tener relaciones semánticas similares (Turney, 2010).

1.1.1.2 Esquemas de pesos

El esquema de pesos más usado es la frecuencia de aparición de los términos en los documentos (*Term Frequency / Inverse Document Frequency*; TF-IDF) para expresar el peso relativo del rasgo o término *w* en el vector asociado a un documento *d* y se calcula según la expresión (1), donde *idf*(*w*) se calcula según la expresión (2).

$$tfidf(w,d) = tf(w,d) * idf(w) \quad (1)$$

[3] Este ejemplo aplica un método conocido por método ventana, el cual es aquel donde las palabras contextuales para una instancia particular son tomadas de una secuencia de palabras que contienen la palabra objetivo. En este caso los límites de las ventanas son las oraciones. Cuando la ventana es tan grande como una oración, un párrafo o un documento, la relación que extrae tiende a ser de similitud tópica. Para sinónimos, por ejemplo, se necesitan contextos más detallados y pequeños (Clark 2014).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

$$idf(w) = log\frac{N}{df(w)} \quad (2)$$

Así, *tf* (*w*,*d*) es la frecuencia del término (cantidad de ocurrencias de la palabra *w* en un documento *d*), *idf* (*w*) es la frecuencia inversa de documentos (cantidad de documentos donde aparece la palabra *w* pero de forma inversa, debido a que se le otorga mayor peso a las palabras que ocurren en una menor cantidad de documentos), *df* (*w*) es la frecuencia de documento (cantidad de documentos que contienen la palabra *w*) y *N* representa la cantidad total de documentos en el corpus (Aggarwal & Zhai, 2012; Manning et al., 2008). La mayoría de las formas de pesado se basa en alguna variación de la fórmula TF-IDF (Manning, 1999; Arco, 2008).

$$w(d,t) = tf_d(t)\left(1+\log_2(\frac{n}{n(t)})\right) \quad (3)$$

Una modificación de la expresión (3) se muestra en (4), teniéndose en cuenta la cantidad de veces que ocurren en un documento aquellos términos que más aparecen, donde $\max_k tf_d(k)$ representa el número de ocurrencias que tiene la palabra que más aparece en *d*.

$$w(d,t) = \frac{tf_d(t)}{\max_k tf_d(k)}\left(1+\log_2(\frac{n}{n(t)})\right) \quad (4)$$

Otra forma para el cálculo de TF-IDF es la expresión (5).

$$w(d,t) = \begin{cases} (1+tf_d(t_i))\log_2(\frac{n}{n(t_i)}), & si\ tf_d(t_i) \geq 1 \\ 0 \quad , & si\ tf_d(t_i) = 0 \end{cases} \quad (5)$$

En (6) se muestra una expresión para el cálculo de TF-IDF que tiene como objetivo obtener pesos en el intervalo [0,1] y considerar la componente de normalización.

$$w(d,t) = \frac{\text{tfidf}(d,t)}{\sqrt{\sum_{j=1}^{m}\left(\text{tfidf}(d,t_j)\right)^2}} \quad (6)$$

La expresión (7) es una variante desglosada de (6), donde el numerador de este coeficiente considera la frecuencia de ocurrencia del término *t* en *d* y la discriminación del término IDF, mientras que el denominador permite la estandarización para eliminar la influencia de la longitud del documento.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

$$w(d,t) = \frac{tf_d(t)\log_2\left(\frac{n}{n(t)}\right)}{\sqrt{\sum_{j=1}^{m}\left(tf_d(t_j)\log_2\left(\frac{n}{n(t_j)}\right)\right)^2}} \quad (7)$$

La información mutua puntual (Pointwise mutual information; PMI) es una medida derivada de la teoría de la información, que vuelve a pesar las ocurrencias usando estadísticas a nivel de corpus para reflejar el significado de las coocurrencias. Ha sido utilizada por ejemplo para medir la orientación semántica de frases (Turney 2001). Dada una palabra *w* y otra palabra *v*, PMI entre *w* y *v* se define en (8) como:

$$\boldsymbol{PMI(w,v) = \log\frac{(p(w,v))}{p(w)p(v)}} \quad (8)$$

donde *p*(*w*, *v*) es la probabilidad de que *w* y *v* coocurran, por ejemplo, en un mismo contexto, y *p*(*w*) y *p*(*v*) son las probabilidades de aparición de las palabras *w* y *v*, respectivamente (Grefenstette, Moritz, et al., 2014) (Curran, 2003).

1.1.1.3 Métodos de reducción de dimensiones

Algunos métodos propuestos para reducir las dimensiones[4] son (Grefenstette, Moritz, et al., 2014):

- LSA: El análisis semántico latente (*Latent Semantic Analysis*; LSA) se propuso por Deerwester y colegas en 1988 (Deerwester, 1988). El modelo representa conceptos semánticos presentes en los documentos. La técnica que emplea este modelo para reducir la dimensionalidad es una técnica de factorización de matrices, llamada descomposición de valores singulares (*Singular Value Descomposition*; SVD) (Deerwester et al., 1990), para encontrar un espacio semántico latente. En esta técnica una matriz de término-documento se descompone[5] en un conjunto de factores ortogonales, a partir de los cuales la matriz original puede aproximarse por una combinación lineal. Si los factores más pequeños son ignorados al multiplicar las matrices más pequeñas se obtiene una aproximación de la matriz de coocurrencia original, este proceso es llamado SVD truncado y es el método que reduce las dimensionalidades en LSA. La idea es que SVD induce relaciones entre filas o entre columnas, que son similares a otras filas o columnas en la matriz de coocurrencia original y de esta forma LSA

[4] Estos métodos también han sido utilizados como formas de representación textual.

[5] La descomposición se basa en la factorización de la matriz principal en tres matrices más pequeñas, el producto de estas matrices (llamadas factores o vectores singulares) da como resultado la matriz original (Sahlgren 2006).

agrupa palabras que ocurren en contextos similares. Se ha destacado que este modelo surge con el objetivo de superar las dificultades semánticas, generadas por la sinonimia y la polisemia (Abella & Medina, 2014). El modelo representa los vectores de documentos en un espacio dimensional asociado a los conceptos presentes en la colección. Por tanto, se considera una forma de representación textual, no obstante, otros autores clasifican este modelo como un método de reducción de dimensionalidad a partir de la representación VSM de un corpus textual (Sahlgren, 2006).

- HAL: El hiperespacio análogo al lenguaje (*Hyperspace analogue to language*; HAL) usa una matriz de coocurrencia palabra-palabra (término-término), la cual contiene coocurrencias de palabras dentro de una ventana de contexto direccional de un tamaño de 10 palabras. Las coocurrencias son pesadas con la distancia entre las palabras, de tal forma que las palabras ocurren próximas unas a otras para obtener el peso mayor, y las palabras que ocurran en lados opuestos de la ventana de contexto obtiene el peso menor. El resultado de esta operación es una matriz de coocurrencia direccional en la que las filas y las columnas representan cantidades de coocurrencia en diferentes direcciones. Cada par fila-columna (es decir, coocurrencias del contexto derecho e izquierdo) son concatenados para producir un vector de contexto de altas dimensiones (tiene dos veces el tamaño del vocabulario). En el caso que manejar estos vectores sea costoso, HAL reduce la dimensionalidad de los mismos al calcular las varianzas de los vectores filas y columna para cada palabra y descartar los elementos con la menor varianza, dejando solo 100 o 200 elementos de los vectores que más varían. A partir de esta representación, se utiliza la medida de Minkowski para calcular la similitud entre vectores (Sahlgren, 2006).

- COALS: es un algoritmo que usa una colección de documentos para construir un espacio semántico, específicamente construye una matriz término-término donde cada elemento en la matriz representa cuan frecuente dos términos ocurren juntos. La matriz es posteriormente normalizada por correlación donde los valores negativos son igualados a cero, y los valores no negativos son reemplazados por su raíz cuadrada. Opcionalmente, la matriz de coocurrencia de términos es reducida con SVD (Rohde et al., 2009).

- Modelos de inducción del sentido de las palabras: Las técnicas de discriminación del sentido de las palabras no supervisadas consisten en agrupar instancias de una palabra objetivo que ocurre en un texto usando espacios vectoriales y valores de similitud. El contexto de cada instancia es representado como un vector en un espacio de rasgos de altas dimensiones. La discriminación se logra al agrupar los vectores contextos directamente en el espacio vectorial y al encontrar valores de similitud entre los vectores y luego realizar un agrupamiento en este

espacio de similitudes. Se emplean dos representaciones distintas del contexto en las que la palabra ocurre, ellas son: representar el contexto de cada instancia de una palabra como un vector de rasgos que ocurren en ese contexto y representar el contexto basado en el promedio de vectores que representan las palabras que ocurren en el contexto (Purandare & Pedersen, 2004).

- RI: La indexación aleatoria (Random indexing; RI) fue desarrollada a partir del trabajo de Kanerva sobre memoria distribuida esparcida (Kanerva, 1988). Se basa fundamentalmente en la idea de acumular vectores de contexto. RI es una técnica que no necesita almacenar una gran matriz de coocurrencia como sucede en LSA y HAL. RI construye los vectores contexto de forma diferente, en vez de almacenar las coocurrencias en una matriz y luego extraer los vectores, RI acumula de forma incremental vectores contexto en dos pasos. Primero, para cada contexto (cada documento o tipo de palabra) se asigna una representación única generada aleatoriamente, llamada vector índice. Estos vectores son dispersos y de alta dimensionalidad (en el orden de miles). Luego, los vectores contexto son acumulados al analizar una palabra a la vez, y se adiciona el o los vectores índices del contexto (los tipos de palabras que están alrededor o los documentos) al vector del contexto de la palabra. Cuando el conjunto de datos completo ha sido procesado, los vectores contexto coinciden con la suma de los contextos de las palabras. Recientemente este modelo fue utilizado para la extracción de sinónimos (Henriksson et al., 2014). RI se caracteriza del resto de los modelos de acuerdo a los siguientes aspectos (Sahlgren, 2006):
    - ✓ Es incremental, significa que los vectores contextos pueden ser usados para cálculos de similitud. Por el contrario, otras implementaciones espacio-palabra requieren que todo el corpus sea muestreado y representado en una matriz de coocurrencia para ejecutar las similitudes.
    - ✓ Utiliza una dimensionalidad fija, lo cual significa que nuevos datos no incrementa la dimensionalidad de los vectores.
    - ✓ Usa reducción de dimensión implícita, debido a que la dimensionalidad fija es menor que la cantidad de contextos en los datos. Esto es ventajoso respecto al consumo de memoria y el tiempo de procesamiento, por lo que es menos costoso que otros métodos.
    - ✓ Es robusto para escoger los parámetros. Las técnicas de proyección aleatoria se ejecutan mejor mientras la dimensionalidad de los vectores es más cercana al tamaño del contexto en los datos.

- NMF: La factorización de matriz no negativa (*Non-Negative Matrix Factorization*; NMF) es un algoritmo de factorización de matrices que se enfoca en el análisis de matrices de datos cuyos elementos son no negativos[6] (Lee et al., 2000). Esta técnica representa una matriz por la factorización de dos matrices. Dado un conjunto de vectores de datos *n*-dimensionales multivariado, los vectores se ubican en las columnas de una matriz X_{nxm} donde *m* es la cantidad de ejemplos del conjunto de datos. Esta matriz es luego factorizada aproximadamente en una matriz W_{nxr} y una matriz H_{rxm}. Usualmente se selecciona *r* para que sea menor que *n* o *m*, de tal forma que *W* y *H* son más pequeñas que la matriz original *X* (Lee et al., 2000). Se puede alcanzar una buena aproximación si los vectores bases descubren una estructura latente u oculta en los datos.

- PLSA: El análisis semántico latente probabilístico (*Probabilistic Latent Semantic Analysis*; PLSA) se propuso en 1999 por Hofmann como una versión probabilística de LSA (Hofmann, 1999). Hofmann declara que LSA posee varias limitaciones debido a que no tiene una base estadística. Por lo que PLSA es un modelo generativo[7] probabilístico, basado en un modelo de aspectos[8] y fue desarrollado para el análisis estadístico del texto. Este modelo se utiliza para descubrir la semántica de tópicos ocultos en documentos usando la representación de bolsa de palabras (Ren & Han, 2014).

- LDA: El modelo de asignación latente de Dirichlet (*Latent Dirichlet Allocation*; LDA) fue introducido por primera vez por David Blei y colegas en el año 2003 (Blei et al., 2003). LDA es un modelo probabilístico generativo para colecciones de datos discretos como es el caso de un corpus textual. Específicamente, LDA es un modelo bayesiano jerárquico de tres niveles (documento, palabra y tópico), el cual considera a un tópico como "*una distribución sobre un vocabulario fijo*" (Blei et al., 2003). El modelo toma previamente una cantidad de tópicos predefinida para toda la colección y se definen las palabras que pertenecen a esos tópicos. El

[6] Matriz de enteros o reales donde cada elemento es un número no negativo (mayor que cero). http://mathworld.wolfram.com/NonnegativeMatrix.html

[7] Un modelo generativo para documentos está basado en reglas simples de muestreo probabilístico que describen cómo las palabras en los documentos pueden estar generadas en las bases de variables latentes (aleatorias). Cuando se adapta un modelo generativo, el objetivo es encontrar el mejor conjunto de variables latentes que pueden explicar los datos observados (es decir, palabras observadas en los documentos), asumiendo que el modelo genera datos realmente (Steyvers & Griffiths 2004).

[8] Un modelo de aspecto es un modelo de mezcla estadístico que está basado en dos suposiciones: primero, se asume que un par de observación documento-palabra (d,w) es generado independientemente (esto corresponde al enfoque de BOW). Segundo, la suposición de independencia condicional, la cual consiste en que la palabra w es generada de forma independiente de un documento d específico, para la clase latente z (Hofmann 1999).

procesamiento del modelo consiste básicamente en identificar en qué medida esos tópicos se presentan en los documentos; primero se escoge una distribución sobre los tópicos; es decir, el conjunto de tópicos predefinidos con sus palabras más probables. Luego, para cada palabra del documento se escoge una asignación de tópicos y se selecciona la palabra para el tópico correspondiente. La salida del proceso de ubicar las palabras por tópicos, los que equivalen a grupos de palabras más frecuentes, y estas palabras son localizadas para cada tópico predefinido más frecuente encontrado en los documentos.

En los últimos años, algunos autores integran nuevas técnicas a los modelos mencionados para mejorar las representaciones textuales basadas en VSM, por ejemplo (Faruqui & Dyer, 2014), (Garrette et al., 2014), (Brychcín & Konopík, 2014), (Jauhar et al., 2015), (Reisinger & Mooney, 2010) así como resaltan la importancia de analizar los distintos parámetros que pueden influir en modelos basados en VSM (Kiela & Clark, 2014).

1.1.1.4 Coeficientes para comparar documentos

Para conocer cuando una unidad textual (palabra, párrafo, oraciones o documentos) es semejante o distinta de otra, la comunidad de investigadores ha presentado varias funciones de similitud, las cuales son útiles en tareas como la recuperación de información, el agrupamiento de documentos, la desambiguación de palabras y la detección de tópicos, entre otras. La similitud entre palabras es fundamental para hallar la similitud de los textos y de esta forma puede ser usada como base para hallar similitudes de otras unidades textuales como oraciones, párrafos y documentos. Las palabras pueden ser similares de forma léxica si tienen secuencias de caracteres similares; y pueden ser similares de forma semántica si tienen el mismo significado, si son usadas en el mismo contexto, en la misma forma o una palabra es un tipo de otra. La similitud léxica se presenta a través de algoritmos basados en cadenas y la similitud semántica a través de algoritmos basados en corpus y en conocimiento. Estos enfoques consisten en (Gomaa, 2013):

- Medidas basadas en cadenas: operan en secuencia de cadenas y composición de caracteres. Miden la similitud o disimilitud entre dos cadenas de textos para compararlas o estimar su correspondencia.
- Medidas basadas en corpus: determinan la similitud entre palabras de acuerdo a la información que se obtiene de grandes corpora.
- Medidas basadas en conocimiento: determinan el grado de similitud entre palabras usando la información obtenida de redes semánticas.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

Las medidas basadas en cadenas han sido clasificadas en medidas de similitud basadas en caracteres y medidas de similitud basadas en términos. Dentro de esta última clasificación se destacan cuatro medidas muy usadas: el coeficiente coseno, el coeficiente Jaccard, el coeficiente Dice y la distancia Euclidiana (Berry & Castellanos, 2007).

La similitud entre dos vectores de términos se puede calcular mediante el ángulo que forman, más concretamente mediante el coseno del ángulo (considerando que cuanto más próximos están dos vectores mayores es la similitud entre ellos). La principal característica de esta medida consiste en realizar una normalización de los vectores de forma más suave, no asignando tanta importancia a los documentos cortos. Cuando el ángulo entre los vectores es menor, la similitud es mayor y en consecuencia el coseno del ángulo es mayor (Seijo et al., 2011). La ecuación (9) representa la medida coseno entre los documentos d_i y d_j, donde d_{ik} representa el peso del rasgo *k* en el documento d_i.

$$sim(d_i, d_j) = cos(\alpha) = \frac{\sum_{k=1}^{m} d_{ik} \cdot d_{jk}}{\sqrt{\sum_{k=1}^{m} d_{ik}^2} \sqrt{\sum_{k=1}^{m} d_{jk}^2}} = \frac{d_i}{|d_i|} \cdot \frac{d_j}{|d_j|} \quad (9)$$

La distancia Euclidiana o distancia L2 mostrada en la ecuación (10) es la raíz cuadrada de la suma de las diferencias cuadradas entre los elementos correspondientes de dos vectores. Esta distancia mide cuán lejos están dos vectores en el espacio vectorial. Esta distancia no arroja buenos resultados cuando la dimensionalidad de los vectores a comparar es alta.

$$dist(d_i, d_j) = \sqrt{\sum_{k=1}^{m} (d_{ik} - d_{jk})^2} \quad (10)$$

El coeficiente Dice se presenta en la ecuación (11), el cual se define como dos veces la cantidad de términos comunes en las cadenas comparadas, divididas por la cantidad total de términos en ambas cadenas.

$$sim(d_i, d_j) = \frac{2(\sum_{k=1}^{m} (d_{ik} \cdot d_{jk}))}{\sum_{k=1}^{m} d_{ik}^2 + \sum_{k=1}^{m} d_{jk}^2} \quad (11)$$

La similitud Jaccard se calcula la cantidad de términos compartidos sobre la cantidad de términos únicos en ambas cadenas. Su fórmula se muestra en la ecuación (12).

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{m} (d_{ik}\, d_{jk})}{\sum_{k=1}^{m} d_{ik}^2 + \sum_{k=1}^{m} d_{jk}^2 - \sum_{k=1}^{m} (d_{ik} d_{jk})} \quad (12)$$

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

Steyvers y Griffiths declaran que la similitud entre dos documentos puede ser medida por la similitud entre sus distribuciones de tópicos correspondientes Θ^{di} y Θ^{dj} (Steyvers & Griffiths 2004). Existen varias funciones de similitud para distribuciones probabilísticas (Steyvers & Griffiths 2004). Una función estándar para medir la diferencia o divergencia entre dos distribuciones d_i y d_j es la divergencia de Kullback Leibler (KL), sus versiones asimétrica y simétrica se presentan las ecuaciones (13) y (14), respectivamente, donde *m* también pudiera representar la cantidad de tópicos que describan los documentos:

$$D(d_i, d_j) = \sum_{k=1}^{m} d_{ik} \, log_2 \frac{d_{ik}}{d_{jk}} \quad (13)$$

$$KL(d_i, d_j) = \frac{1}{2}\left[D(d_i, d_j) + D(d_j, d_i)\right] \quad (14)$$

Otra opción es aplicar la divergencia simetrizada de Jensen-Shannon (JS), la cual se expone en la ecuación (15) y mide la similitud entre *p* y *q* a través del promedio de d_i y d_j. Dos distribuciones d_i y d_j serán similares si son similares a su promedio $(d_i + d_j)/2$.

$$JS(d_i, d_j) = \frac{1}{2}\left[D(d_i, (d_i + d_j)/2) + D(d_j, (d_i + d_j)/2)\right] \quad (15)$$

Se pueden considerar las distribuciones de tópicos como vectores y aplicar funciones motivadas geométricamente como la distancia Euclidiana y la similitud coseno. En modelos probabilísticos la similitud entre dos palabras puede medirse por el alcance que comparten los mismos tópicos, obteniéndose las distribuciones de tópicos condicionales $\theta^{(1)}$ y $\theta^{(2)}$ mostradas en (16) y (17) respectivamente; w_1 y w_2 representan las palabras y *z* el tópico.

$$\theta^{(1)} = P(z|w_i = w_1) \quad (16)$$

$$\theta^{(2)} = P(z|w_i = w_2) \quad (17)$$

Las funciones de KL o JS pueden usarse para medir la similitud distribucional entre estas distribuciones. Cualquiera que sea la función de similitud o relevancia usada, requiere obtener estimaciones estables para las distribuciones de tópicos, fundamentalmente para documentos pequeños.

1.1.2 Modelos basados en la predicción de contextos

Una tarea que ha sido analizada recientemente para la construcción de modelos de vectores de significado es aprender representaciones de vectores para palabras (Grefenstette, Moritz, et al., 2014).

Varias investigaciones han dedicado sus esfuerzos a representar vectores de aprendizaje de palabras usando redes neuronales (Bengio et al., 2003; Mikolov, 2013). La idea es que cada palabra se representa por un vector que es concatenado o promediado con vectores palabras en un contexto, y el vector resultante es usado para predecir otras

palabras en el contexto. Por ejemplo, un modelo de lenguaje de red neuronal propuesto en (Bengio et al., 2003) usa la concatenación de varios vectores palabra anteriores para formar la entrada de una red neuronal, y trata de predecir la próxima palabra. Luego que el modelo es entrenado, la salida consiste en hacer corresponder los vectores palabra a espacios vectoriales de tal forma que las palabras semánticamente similares tienen representaciones vectoriales similares (por ejemplo, "fuerte" está cerca de "poderoso") (Mikolov & Com, 2014).

Una diferencia importante respecto a las representaciones espacio vectorial para las palabras, es que las representaciones neuronales buscan representaciones para palabras que sean útiles para representar la distribución de probabilidad de secuencias de palabras del texto en lenguaje natural de forma compacta. Esta propuesta intenta combatir el problema de la dimensionalidad (*curse of dimensionality*), en el cual una secuencia de palabras en la que el modelo será probado es probable que sea diferente de todas las secuencias que fueron vistas durante el entrenamiento. La propuesta aprende una representación distribuida para las palabras que permite a cada oración de entrenamiento informar el modelo sobre una cantidad exponencial de oraciones vecinas semánticamente. Los pasos básicos propuestos por (Bengio et al., 2003) para aprender el modelo de red neuronal son:

- ✓ Asociar con cada palabra en el vocabulario un vector de rasgos de palabra distribuido (con valores reales).
- ✓ Expresar una función de probabilidad unificada (*joint probability function*) de secuencias de palabras en términos de los vectores de rasgos de estas palabras en la secuencia.
- ✓ Aprender vectores de rasgos de palabra y los parámetros de esa función de probabilidad.

El vector de rasgos representa diferentes aspectos de una palabra, cada palabra está asociada a un punto en el espacio vectorial. La función de probabilidad se expresa como el producto de probabilidades condicionales de la próxima palabra dada las anteriores, por ejemplo, usando una red neuronal multicapa para predecir la próxima palabra. El modelo generaliza combinaciones porque se espera que palabras similares tengan vectores de rasgos similares. Un ejemplo es "gato" y "perro", estas palabras presentan roles semánticos y sintácticos similares en las oraciones "El gato está caminando en el cuarto" y "El perro estaba corriendo en el patio". A partir de estas oraciones se pueden generalizar otras combinaciones como "El gato corre por el patio" y "El perro estaba caminando en el cuarto" (Bengio et al., 2003).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

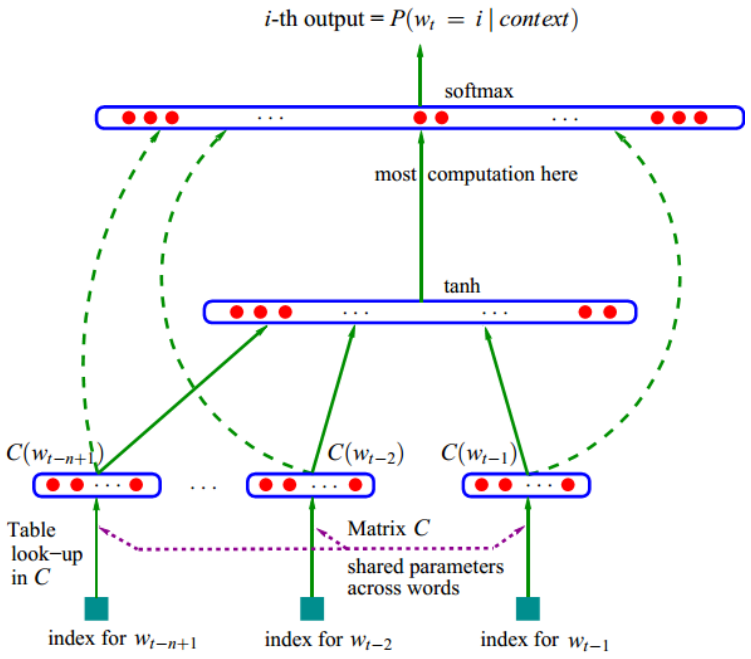

Figura 3. Arquitectura neuronal propuesta por (Bengio et al. 2003).

En la Figura 3 se muestra una arquitectura neuronal para un conjunto de entrenamiento representado por una secuencia $w_1 \ldots w_T$ de palabras $w_t \in V$ para el vocabulario V (el cual es un conjunto finito y grande). El objetivo es aprender un modelo $f(w_t, \ldots, w_{t-n+1}) = \hat{P}(w_t | w_1^{t-1})$ que da una alta probabilidad; esta arquitectura descompone el modelo en dos partes. Primero, un mapeo de C para cualquier elemento i de V para un vector real $C(i)$, el cual representa vectores de rasgos distribuidos asociados con cada palabra en el vocabulario. C se representa por una matriz $|V| \times m$ de parámetros libres. Segundo, la función de probabilidad sobre palabras expresada con C: una función g mapea una secuencia de entrada de vectores de rasgos para palabras en un contexto $(C(w_t - n+1), \cdots, C(w_t - 1))$, a una distribución de probabilidad condicional sobre palabras en V para la próxima palabra w_t. La salida de g es un vector cuyo elemento i estima la probabilidad $\hat{P}(w_t = i \mid w_1^{t-1})$ expresado como se muestra en la ecuación , donde g es una red neuronal y $C(i)$ es el vector de rasgos de palabra (Bengio et al. 2003).

$$f(i, w_{t-1}, \ldots, w_{t-n+1}) = g\left(i, C\left(w_{t-1}, \ldots, C(w_{t-n+1})\right)\right) \quad (18)$$

Recientemente, se presentaron dos arquitecturas basadas en redes neuronales para aprender representaciones de vectores que capturan relaciones semánticas y sintácticas de palabras (Mikolov, Corrado, et al. 2013). Fueron nombradas modelo de Bolsa de Palabras Continuo (*Continuos Bag of Words*; CBOW) y modelo Skip-gram. CBOW presenta una

arquitectura similar a un modelo de lenguaje de red neuronal *feedforward*, con capas de entrada, de proyección, ocultas y de salida para predecir la palabra actual basada en el contexto. Sus vectores son promediados y el orden de las palabras no influye en la proyección. Usan un clasificador log-lineal para clasificar las palabras. *Skip-gram* a diferencia de CBOW, trata de maximizar la clasificación de una palabra basada en otra palabra en la misma oración. Para ello usa la palabra actual como entrada para un clasificador log-lineal con una capa de proyección continua y predice palabras dentro de un rango antes y después de la palabra actual.

Recientemente, algunos autores han realizado estudios de comparación entre los enfoques de modelos de conteo y los modelos predictivos (Baroni et al., 2014). En sus experimentaciones obtuvieron mejores resultados con los modelos predictivos. Argumentan que los pesos de los vectores son establecidos directamente para predecir de forma óptima los contextos en los que las palabras correspondientes tienden a aparecer y debido a que palabras similares ocurren en contextos similares el sistema aprende a asignar vectores similares a palabras similares. Declaran que esta nueva forma de entrenar los modelos semánticos distribucionales es atractiva porque reemplaza el cálculo heurístico de las transformaciones de vectores de los modelos iniciales, con un paso de aprendizaje supervisado. La supervisión no tiene un costo de anotación manual, dado que la ventana de contexto usada para entrenar puede ser extraída automáticamente de un corpus no anotado. Sin embargo, este enfoque es dependiente de la calidad del corpus original y del dominio.

1.2 Semántica composicional

Otra tarea analizada recientemente para la construcción de modelos de vectores de significado es aprender como componer los modelos para obtener representaciones de vectores para frases, oraciones y documentos (Grefenstette, Moritz, et al., 2014) (Grefenstette, Sadrzadeh, et al., 2014; Krishnamurthy et al., 2013; Hermann & Blunsom, 2013; Thater et al., 2010).

La representación distribucional permite conocer si dos palabras significan lo mismo aproximadamente, dada una representación vectorial se utiliza algún coeficiente para calcular la similitud entre las unidades. Sin embargo, para conocer si dos oraciones significan lo mismo no se puede usar el mismo enfoque, porque no se pueden aprender rasgos distribucionales a nivel de oración. Desde el punto de vista lingüístico el lenguaje se entiende a través de unidades compuestas, como palabras y frases, pero no memorizando oraciones. La composición semántica permite aprender una jerarquía de rasgos, donde niveles más altos de abstracción son derivados a partir de niveles más bajos (Grefenstette, Moritz, et al., 2014). Una función genérica de composición semántica puede expresarse como $p = f(u, v, R, K)$ donde u y v son representaciones hijas, R es la información relacional y K el conocimiento histórico (*background knowledge*).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

Las estructuras lingüísticas son compuestas, donde elementos más simples forman elementos más complejos. Por ejemplo, los morfemas forman palabras, las palabras forman frases, y las frases forman oraciones. Sin embargo, la similitud semántica es más compleja que una relación simple entre palabras independientes. El contenido semántico de una oración está relacionado al contenido de sus constituyentes y la habilidad de recombinarlo de acuerdo a un conjunto de reglas. Las redes neuronales pueden representar objetos individuales distintos, pero en el caso de múltiples objetos existen dificultades en conocer cuáles rasgos están relacionados a determinados objetos; por ejemplo, no se han logrado buenas representaciones de oraciones como "José ama a María" y "María ama a José". Igualmente, los modelos basados en semántica distribucional no son efectivos para representar relaciones composicionales, porque la representación semántica de estos modelos está enfocada a palabras individuales. Por ejemplo, enfoques como las matrices de par-patrón no son composicionales, capturan el significado de pares de palabras y frases como un todo, sin modelar las partes que lo constituyen (Mitchell & Lapata, 2010).

Una propuesta para modelar frases y oraciones basadas en vectores utiliza una función de dos vectores y presenta modelos basados en operaciones de adición y multiplicación. Su enfoque responde al problema de combinar vectores semánticos para hacer una representación de frases multipalabras, el cual es diferente al problema de como incorporar información sobre contextos multipalabras en representaciones distribucionales para una palabra individual. En esta propuesta se simplifica la función genérica $p = f(u, v)$, la cual se utiliza con la idea que una oración típicamente consiste de operaciones de composición, cada una se aplica a un par de constituyentes *u* y *v*. Por ejemplo, un modelo aditivo de composición es $p = Au + Bv$ donde *A* y *B* son matrices que determinan las contribuciones hechas por *u* y *v* a p. Una función de composición simple dentro de este modelo es $p = u + v$. Una variante de un modelo representa la composición en términos de la suma de predicado, argumento y cantidad de vecinos del predicado de la siguiente forma $p = u + v + \sum_i n_i$. Una función de multiplicación es $p = u \odot v$ donde $\odot$ indica la multiplicación de los componentes correspondientes: $p_i = u_i \, . \, v_i$ para obtener. Sin embargo, estas funciones no tienen en cuenta el orden de las palabras, ni la sintaxis, ni relaciones gramaticales (Mitchell & Lapata 2010). Un ejemplo que permite ilustrar las operaciones composicionales descritas puede desarrollarse a partir de la frase "practical difficulty", donde *u* representa "practical" y *v* representa "difficulty". Los vectores hipotéticos para estos componentes se muestran en la Figura 4. La operación de adición en este caso equivale a *practical* + *difficulty* = [1 14 6 14 4] y la operación de multitplicación *practical* $\odot$ *difficulty* = [0 48 8 40 0]. La operación que tiene en cuenta la cantidad de vecinos del predicado sería *practical* + *difficulty* + *problem* = [3 29 13 23 5], considerando a *problem* = [2 15 7 9 1] como el vector vecino. En (Blacoe & Lapata 2012) se muestran experimentos más recientes para hallar similitud entre frases y detectar paráfrasis.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

	music	solution	economy	craft	reasonable
practical	0	6	2	10	4
difficulty	1	8	4	4	0

Figura 4. Un espacio semántico hipotético para "*practical*" y "*difficulty*" (Mitchell & Lapata 2010).

Con el objetivo de representar estas relaciones semánticas en las oraciones se han propuesto los modelos de función léxica, en los que, por ejemplo, se aprenden matrices de adjetivos (Baroni & Zamparelli, 2010), donde el adjetivo es una función lineal de un vector a otro vector; el primero representa un sustantivo y el segundo vector representa una composición de adjetivo-sustantivo. La regresión lineal es utilizada para aprender un mapa lineal para un adjetivo específico, aplicado a pares compuestos por un sustantivo y vectores adjetivo-sustantivo de un corpus. En este método de mapeo lineal para un adjetivo se realiza una multiplicación de una matriz (que representa el peso del adjetivo) con un vector columna (que representa el sustantivo). Los pasos de este método se pueden resumir de la siguiente forma (Grefenstette, Moritz, et al., 2014):

1. Obtener un vector para cada sustantivo en el diccionario (lexicon).
2. Almacenar pares de sustantivo-adjetivo a partir de un corpus.
3. Obtener vectores de cada bigrama adjetivo-sustantivo.
4. Formar el conjunto de tuplas que representan relaciones de sustantivos a cada adjetivo identificado.
5. Aplicar método de regresión lineal.

Aunque usan como método de aprendizaje supervisado la regresión de mínimos cuadrados, no utilizan datos manualmente anotados, debido a que los vectores son automáticamente almacenados del corpus. Los modelos de función léxica han sido aplicados generalmente a frases cortas o tipos particulares de composición, por ejemplo los sustantivos compuestos. Para representar relaciones entre verbos y adverbios se han estudiado también los tensores[9], con el objetivo de insertar aspectos lógicos en modelos distribucionales de semántica, específicamente la modelación de valores verdaderos, el dominio lógico y sus elementos, predicados y relaciones (Grefenstette, 2013). No obstante,

[9] los tensores son objetos matemáticos que se encuentran en el área del álgebra multi-lineal; pueden ser considerados como generalizaciones de vectores y matrices. Intuitivamente, una estructura de datos vectorial es llamada tensor de orden 1 y una matriz es un tensor de orden 2; un cubo es una estructura de tensor de orden 3 (Liu et al. 2005).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

estas representaciones tienen como desventaja que son difíciles de aprender y no son eficientes para el gran cúmulo textual que existe hoy en día (Grefenstette, Moritz, et al., 2014).

De esta forma propuestas más recientes están guiadas por estudios sobre redes neuronales recurrentes (Socher et al., 2012; Chen et al., 2013; Zou et al., 2012; Socher et al., 2014). En la Figura 5 se muestra una red neuronal recurrente, donde cada palabra y frase están representadas por un vector y una matriz, respectivamente; por ejemplo very=(a, A). La matriz es aplicada a vectores vecinos. La misma función se repite para combinar la frase "*very good*" con "*movie*".

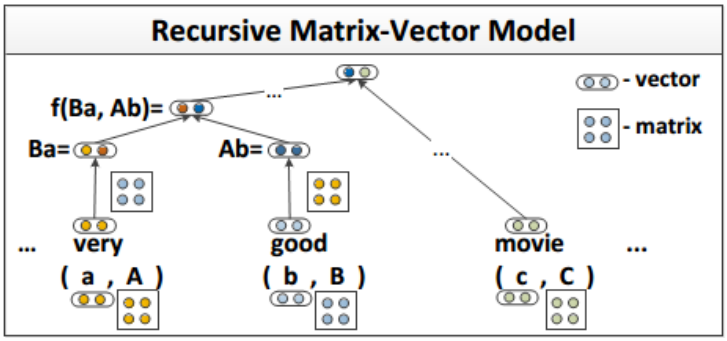

Figura 5. Red neuronal recursiva que aprende representaciones vectoriales semánticas de frases en una estructura de árbol. Fuente: (Socher et al. 2012).

Este enfoque presenta un modelo de red neuronal recursiva que aprende representaciones de vectores composicionales para frases y oraciones de tamaño y sintaxis arbitrarios. El modelo asigna un vector y una matriz a cada nodo en un árbol gramatical, el vector calcula el significado de cada elemento (una palabra o frase larga), mientras que la matriz indica cómo cambia el significado de palabras o frases vecinas. Esta propuesta aprende el significado de operadores en la lógica preposicional y el lenguaje natural, predice sentimientos y puede ser usada para clasificar relaciones semánticas entre sustantivos en una oración.

La representación del modelo está compuesta por una palabra como un vector continuo y una matriz de parámetros. Se inicializan todos los vectores de palabra a partir de un modelo no supervisado que posee vectores de palabras pre-entrenados. Utilizan textos de Wikipedia para que el modelo aprenda vectores de palabra al predecir cuán probable es que una palabra ocurra en su contexto; esto tiene como desventaja que depende de la calidad y diversidad del texto que se utiliza para calcular las probabilidades. Similar a los modelos espacio vectorial basados en coocurrencia, los vectores resultantes capturan información sintáctica y semántica. Luego, cada palabra se asocia a una matriz. Así, representan cualquier frase u oración de tamaño *m* como una lista ordenada de pares de vectores matrices ((*a*, *A*), ..., (*m*, *M*)).

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

Para modelar la composición entre dos palabras se define la función: $p = f_{A,B}(a,b) = f(Ba,Ab) = g\,(W \begin{bmatrix} Ba \\ Ab \end{bmatrix})$ donde *A* y *B* son matrices para palabras únicas, *W* es una matriz que mapea las palabras transformadas en el mismo espacio *n*-dimensional. Para la función *g* utilizan una función no lineal sigmoidea o tangente hiperbólica. A partir de los vectores se obtiene una red neuronal individual $p = g(W z)$, donde *z* es un vector. De esta forma las matrices capturan efectos composicionales para cada palabra.

Posteriormente, se extiende el modelo composicional para aprender vectores y matrices de secuencias más largas (frases). En esencia, se aplica la función *f* a pares de constituyentes en un árbol gramatical; *f* puede ser usada para vectores frase de forma recursiva, por lo que para matrices frase se define: $P = f_M(A,B) = W_M \begin{bmatrix} A \\ B \end{bmatrix}$. Una vez modelado un constituyente (palabra) en el árbol gramatical, este puede mezclarse con otro al aplicar las mismas funciones. El modelo calcula los vectores y matrices de un modo de abajo hacia arriba aplicando las funciones *f* y f_M de forma recursiva con su propia salida anterior hasta que alcanza el nodo superior del árbol el cual representa la oración completa (Socher et al., 2012).

Una propuesta no supervisada para aprender representaciones de rasgos de tamaño fijo a partir de piezas de tamaño variable de texto, como oraciones, párrafos y documentos fue realizada por (Mikolov & Com, 2014). El algoritmo representa cada documento como un vector denso que es entrenado para predecir palabras en el documento. La propuesta concatena un vector de párrafo con varios vectores palabra a partir de un párrafo, y predice la siguiente palabra en un contexto dado. El algoritmo presenta dos etapas, una de entrenamiento (para obtener los vectores palabra) y otra de inferencia (para obtener vectores párrafos). Los vectores son aprendidos a partir de datos no etiquetados.

La composición semántica se ha declarado como la tarea de entender el significado del texto al componer los significados de palabras individuales y la descomposición semántica como la tarea de entender el significado de una palabra individual al descomponerla en varios aspectos que están ocultos en el significado de la palabra. Resultados publicados en (Turney 2014) se enfocan en cómo generar composiciones en vez de como reconocerlas, específicamente sobre unigramas y bigramas sustantivos; considerando un enfoque distribucional en el que una palabra es representada por un vector de contexto. Por ejemplo, un bigrama "azúcar leche" obtenido de la frase "azúcar de la leche" está compuesto por el sustantivo "azúcar" y el sustantivo o adjetivo "leche" que modifica el significado de "azúcar". Dados vectores de contexto para el sustantivo y el modificador, el objetivo es modelar el significado del bigrama. Una prueba para este modelo es que pueda reconocer cuando el unigrama "lactosa" es sinónimo de "azúcar leche". En la investigación se generan listas ordenadas en dos etapas: utilizan dos algoritmos para generar listas iniciales de

candidatos de forma no supervisada, y un tercer algoritmo supervisado que refina la lista al utilizar conjuntos de datos de entrenamiento para construir modelos para las tareas de composición y descomposición. De esta forma, una prueba de composición semántica es que dados vectores de contexto para el sustantivo y un modificador en un bigrama sustantivo-modificador, por ejemplo en Inglés "red salmon", generar un unigrama sustantivo que sea sinónimo de un bigrama como "sockeye". En el caso de la descomposición semántica, dado un vector contexto para un unigrama sustantivo "copa" se desea generar un bigrama sustantivo-modificador que sea sinónimo para un unigrama "vaso de brandy" (Turney, 2014). Otro trabajo interesante en este contexto es (Turney, 2013).

2. Herramientas computacionales

Existen varias herramientas para el procesamiento del lenguaje natural. A continuación se describen aquellas que incluyen implementaciones de los algoritmos ideados para los modelos espacio vectorial que incorporan elementos semánticos, tanto con enfoque distribucional como composicional.

2.1 SemanticVectors

La biblioteca de código abierto y libre nombrada *SemanticVectors*[10] crea modelos de espacio palabra (*word space models*) a partir de texto en lenguaje natural. Los modelos son diseñados para representar palabras y documentos basados en conceptos. Pueden ser usados para tareas como generación automática de tesauros, representación de conocimiento y encontrar términos o conceptos relacionados a un término en específico. Puede trabajar con tres tipos de vectores: reales, complejos y binarios. Los modelos se crean al aplicar algoritmos de conceptos a matrices término-documento creadas con Apache Lucene. Los algoritmos implementados son: *Random Projection*[11], LSA y *Reflective Random Indexing* (RRI). Sus autores declaran que *Random Projection* es la técnica más escalable en la práctica porque no utiliza algoritmos de descomposición de matrices costosos computacionalmente. El algoritmo *Reflective Random Indexing*[12] está basado en *Random Projection*, el cual realiza el proceso de entrenar un modelo semántico en varias fases. La forma básica en la que crean los modelos sigue tres pasos:

1. Crear vectores aleatorios básicos para cada documento.
2. Crear vectores de términos al sumar los vectores de documentos básicos donde el término ocurre.

[10] https://github.com/semanticvectors/semanticvectors/wiki

[11] Se refiere al algoritmo de reducción de dimensiones Random Indexing (Sahlgren 2004)

[12] http://www.j-biomed-inform.com/article/S1532-0464(09)00120-8/fulltext

3. Crear nuevos vectores de documentos al sumar los vectores de términos de los términos que ocurren en cada documento.

La idea de realizar ciclos de entrenamiento es que la salida de la etapa 3 puede ser una entrada en la etapa 2, debido a que pueden usar los vectores de documentos como los vectores de documentos básicos para calcular vectores de términos. RRI es capaz de encontrar conexiones significativas entre los términos que no coocurren juntos en cualquier documento del corpus, puede ejecutarse de dos formas, basado en términos (TRRI) de tal forma que un conjunto de vectores elementales aleatorios se crean para cada término o basado en documento (DRRI), para el cual el punto de entrada es un conjunto de vectores de documentos aleatorios (Widdows & Cohen, 2010). También posee una implementación del algoritmo HAL. Para los resultados de búsqueda utiliza una implementación del algoritmo de agrupamiento K-means (Aggarwal & Zhai, 2012).

2.2 S-Space Package

S-Space Package[13] es una biblioteca de código abierto y libre para desarrollar y evaluar algoritmos de espacio palabra. Los algoritmos son divididos en cuatro categorías basándose en su similitud estructural:

- Modelos basados en documentos: dividen el corpus en documentos discretos y construyen un VSM a partir de las frecuencias de las palabras en los documentos. Por ejemplo: VSM, LSA.
- Modelos basados en coocurrencia: construyen el espacio vectorial usando la distribución de palabras coocurrentes en un contexto, el cual puede ser definido como una región alrededor de una palabra o caminos en un árbol gramatical. Por ejemplo: HAL, COALS (Rohde et al., 2009).
- Modelos basados en aproximación: aproximan datos de coocurrencia para lograr mejor escalabilidad de grandes conjuntos de datos. Por ejemplo: Random Indexing y RRI (Cohen et al., 2010).
- Modelos basados en inducción del sentido de las palabras: intentan descubrir sentidos diferentes de las palabras mientras construyen un espacio vectorial. Por ejemplo: Purandare and Pedersen (Purandare & Pedersen, 2004).

En esencia, la idea de estos modelos es que los rasgos de palabras se extraen de un corpus y la distribución de estos rasgos es usada como base para la similitud semántica. Para las matrices se utilizan esquemas de peso como TF-IDF y PMI. Poseen, además, algoritmos de agrupamiento de tipo aglomerativo jerárquico, agrupamiento espectral y es posible la integración con la biblioteca CLUTO[14]. Algunas de las medidas de similitud que posee son la medida coseno, Euclidiana, Jaccard y KL divergence (Jurgens & Stevens, 2010).

[13] https://github.com/fozziethebeat/S-Space/wiki/GettingStarted

[14] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

2.3 Word2Vector

Word2Vector[15] es una biblioteca de código abierto y libre que provee una implementación de las arquitecturas CBOW y *Skip-gram* para calcular representaciones vectoriales de palabras. La herramienta tiene como entrada un corpus textual y produce los vectores palabra como salida. Primero construye un vocabulario del texto de entrenamiento y luego aprende representaciones vectoriales de palabras. Una forma simple de verificar las representaciones aprendidas es encontrar las palabras más cercanas para una palabra especificada por el usuario. Para observar regularidades fuertes en el espacio vectorial de palabras es necesario entrenar los modelos en grandes conjuntos de datos (desde cientos a billones de palabras). Además, se pueden obtener clases de palabras de grandes conjuntos de datos, para ello utilizan el algoritmo de agrupamiento K-means sobre los vectores palabras. Para el entrenamiento se debe tener en cuenta que:

- La arquitectura Skip-gram es más lenta que CBOW, aunque es buena para capturar palabras poco frecuentes.
- El algoritmo de entrenamiento *hierarchical softmax* es bueno para palabras poco frecuentes y el *negative sampling* es mejor para palabras frecuentes en vectores de baja dimensión.
- Usualmente es mejor mayor dimensionalidad para los vectores palabras, pero no siempre.
- El tamaño de la ventana de contexto a utilizar al aplicar la arquitectura Skip-gram es 10, y el tamaño de ventana recomendado para aplicar CBOW es 5.

Esta biblioteca tiene implementaciones en Java, Python y C (Mikolov, Chen, et al., 2013).

2.4 GloVe

Existen dos familias de modelos para aprender vectores de palabras: los métodos de factorización de matrices globales como LSA, y los métodos basados en ventanas de contexto local como el modelo *Skip-gram*. Algunas desventajas mencionadas sobre estos modelos es que por ejemplo en el caso de LSA no tiene en cuenta la analogía de palabras aunque obtiene información estadística de forma eficiente. En el caso de *Skip-gram* sus esfuerzos son mejores para la analogía de palabras pero casi no utiliza las estadísticas del corpus debido a que entrenan ventanas de contexto locales de forma independiente en vez de hacerlo en frecuencias de coocurrencia globales (Pennington et al., 2014).

Una propuesta de código abierto para representar un espacio vectorial de palabra con sub-estructuras significativas es GloVe[16] (*Global Vectors*), el cual captura estadísticas del corpus global de forma directa. Utilizan un modelo de

[15] http://code.google.com/p/word2vec/

[16] http://nlp.stanford.edu/projects/glove/

mínimos cuadrados (modelo de regresión *log-bilineal global*) que realiza un entrenamiento sobre una matriz de coocurrencia palabra-palabra. GloVe está implementado en C y tiene un enfoque de aprendizaje no supervisado.

El objetivo de entrenamiento de GloVe es aprender vectores palabras de tal forma que el producto escalar (*dot product*) sea igual al logaritmo de la probabilidad de coocurrencia de las palabras. El modelo es entrenado con las entradas distintas de cero de una matriz de coocurrencia global palabra-palabra, la cual contiene cuán frecuentemente coocurren las palabras entre ellas en un corpus. Esta representación requiere una sola pasada por todo el corpus para obtener las estadísticas. Para grandes corpus es costoso, pero se ejecuta una sola vez.

2.5 Gensim

Gensim[17] es una biblioteca de código abierto y libre implementada en Python, que posee implementaciones de los algoritmos no supervisados como LSA y *Random Projection* para descubrir estructuras semánticas en documentos textuales y detecta tópicos con LDA. Presenta varios esquemas de peso como TF-IDF y posee compatibilidad con las bibliotecas de NumPy y SciPy. Permite usar el paradigma de computación distribuida para LSA y LDA y así acelerar los cálculos.

2.6 Dissect

La herramienta de composición semántica distribucional (*DIStributional SEmantics Composition Toolkit; DISSECT*[18]) forma parte del proyecto de operaciones composicionales en el espacio semántico (*COMPositional Operations in SEmantic Space; COMPOSES*); está implementada en Python y es de código abierto. Puede construir espacios semánticos a partir de matrices de coocurrencias, realizar operaciones composicionales y medir similitud semántica entre palabras y frases. Esta herramienta, para la creación de espacios semánticos con matrices de coocurrencia, utiliza dos pasos fundamentales: preprocesamiento del corpus para almacenar las cantidades relevantes y el procesamiento matemático de las cantidades extraídas. Esta herramienta no soporta pre-procesamiento o conteo, pero toma como entrada de forma directa una matriz de coocurrencia. Es decir, que DISSECT se enfoca en los métodos de repesado como PMI y métodos de reducción de dimensiones como SVD y NMF. El principal propósito de DISSECT es incorporar las funciones de composición de vectores que se ha, propuesto en la literatura, por ejemplo, los modelos propuestos en (Mitchell & Lapata, 2010) (Baroni & Zamparelli, 2010) y otros. Algunos modelos son el modelo aditivo pesado, modelo de dilatación (*dilation*), modelo completamente aditivo, modelo de composición de función léxica.

[17] http://radimrehurek.com/gensim/

[18] http://clic.cimec.unitn.it/composes/toolkit/

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 148-180

Tabla 1. Herramientas que realizan análisis semántico de los textos a procesar.

Herramientas	Lenguaje	Semántica distribucional (palabra)		Semántica composicional (frases, oraciones, párrafos y documentos)
		Vectores de conteo	Predicción del contexto	
SemanticVectors	Java	LSA, RI, RRI, HAL y presenta esquemas de peso como TF-IDF y realiza factorizaciones de matrices (SVD)		
S-Space Package (Presenta integración con Word2vector, GloVe)	Java	LSA, HAL, COALS, RI, RRI y presenta esquemas de pesos (PMI) y realiza factorizaciones de matrices (SVD, NMF)		
Word2Vector	Java, Python, Spark MLib		Modelos Skip-gram y CBOW basados en redes neuronales	Permite representar frases
GloVe	C, Java		Modelo de mínimos cuadrados (modelo de regresión log-bilineal global)	
Gensim (Presenta integración con Word2vector)	Python	LSA, RI, LDA Presenta esquemas de peso como TF-IDF		Permite representar frases, oraciones, párrafos y documentos
Dissect	Python	Presenta esquemas de pesos (PMI) y realiza factorizaciones de matrices (SVD y NMF)		Operaciones vectoriales propuestas por (Mitchell & Lapata 2010) para frases

Finalmente, se muestra la Tabla 1 que resumen las herramientas descritas y sus características principales que le permiten realizar representaciones textuales incluyendo elementos semánticos. Aquí hemos mostrado las principales herramientas que incorporan elementos semánticos en las representaciones textuales, no obstante, otras herramientas con tales propósitos se describen en (Jurgens & Pilehvar, 2015).

Conclusiones

Varias son las tareas de NLP que requieren la incorporación de semántica en la representación textual. Ejemplo de ellas son la clasificación de documentos, el análisis de sentimiento, la modelación de lenguaje, la detección de paráfrasis, la traducción automática, la extracción de información, los sistemas de preguntas y respuestas, etc. La representación textual con modelos semánticos distribucionales, específicamente los que aplican el cálculo de la coocurrencia de las palabras en los documentos ha solucionado una variedad de problemas en áreas como la recuperación de información y el aprendizaje automático. Por ejemplo, la mayoría de los motores de búsqueda que se encuentran en internet utilizan

matrices del tipo término-documento y palabra-contexto, y aplican técnicas de reducción de dimensiones como las citadas en la presente revisión. No obstante, debido a la inmensa cantidad de información que se genera cada minuto y a la influencia que tienen las tecnologías informáticas, la internet de las cosas y la computación móvil en la vida humana, actualmente se hace necesario que las herramientas computacionales permitan comunicar la información de forma más entendible para las personas. Las representaciones de textos en espacios vectoriales semánticos contribuyen a dar respuesta a esta necesidad y su estudio seguirá en ascenso para mejorar las herramientas del futuro.

Los modelos espacio vectorial de significado han probado ser efectivos en el campo de la lingüística computacional y su estudio es un área activa de investigación actualmente. Dentro de ellos, los modelos de semántica distribucional son útiles para representar el significado de palabras individuales pero no para representar relaciones composicionales presentes en frases, oraciones y párrafos. Las técnicas de aprendizaje con redes neuronales son efectivas y muy investigadas en la actualidad para aprender representaciones de vectores de forma sintáctica y semántica con palabras, frases, oraciones y documentos. Además, existe una variedad de herramientas computacionales de código abierto que implementan estos modelos, fundamentalmente en lenguaje Java y Python.

Referencias

ABELLA, R. & MEDINA, J. Segmentación lineal de texto por tópico. Serie Gris CENATAV. 2014

AGGARWAL, C.C. & ZHAI, C. Mining Text Data, Springer. 2012

ARCO, L. Agrupamiento basado en la intermediación diferencial y su valoración utilizando la teoría de los conjuntos aproximados. UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS. 2008

BAEZA-YATES, R. & RIBEIRO-NETO, B. Modern Information Retrieval, 1998

BARONI, M., DINU, G. & KRUSZEWSKI, G. Don’t count, predict! Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014. pp.238–247.

BARONI, M. & ZAMPARELLI, R. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. EMNLP. 2010

BENGIO, Y. ET AL. A Neural Probabilistic Language Model. Journal of Machine Learning Research, 3, pp.1137–1155. 2003

BERRY, M.W. & CASTELLANOS, M. Survey of Text Mining: Clustering, Classification, and Retrieval, Second Edition. 2007

BIGGS, N.; LLOYD, E. WILSON, R. Graph Theory, Oxford University Press. 1986

BLACOE, W. & LAPATA, M. A Comparison of Vector-based Representations for Semantic Composition. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, (July), pp.546–556. 2012

BLEI, D.M., NG, A.Y. & JORDAN, M.I. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 2003. pp.993–1022.

BRYCHCIN, T. & KONOPIK, M. Semantic spaces for improving language modeling. Computer Speech & Language, 28(1), 2014. pp.192–209. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0885230813000387.

CHEN, D. et al. Neural Tensor Networks and Semantic Word Vectors. Advances in Neural Information Processing Systems, 26, 2013. pp.1–4.

CLARK, S. Vector Space Models of Lexical Meaning. In S. Lappin & C. Fox, eds. Handbook of Contemporary Semantics. 2014. pp. 1–43.

COHEN, T., ROGER, S. & WIDDOWS, D. Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. Journal of Biomedical Informatics, 43(2), 2010. pp.240–256.

CURRAN, J.R. From Distributional to Semantic Similarity. 2003

DEERWESTER, S. Improving Information Retrieval with Latent Semantic Indexing. Proceedings of the 51st Annual Meeting of the American Society for Information Science, 1988. pp.36–40.

DEERWESTER, S. ET AL. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), 1990. pp.391–407.

FARUQUI, M. & DYER, C. Improving Vector Space Word Representations Using Multilingual Correlation. Association for Computational Linguistics, 2014. pp.462–471.

FIRTH, J.R. A synopsis of linguistic theory 1930-1955. Studies in linguistic analysis. 1957. pp.1–32.

GARRETE, D., ERK, K. & MOONEY, R. A Formal Approach to Linking Logical Form and Vector-Space Lexical Semantics. Computing Meaning, 2014. pp.27–28.

GOMAA, W.H. A Survey of Text Similarity Approaches. International Journal of Computer Applications, 68(13), 2013. pp.13–18.

GREFENSTETTE, E., SADRZADEH, M., et al. Concrete Sentence Spaces for Compositional Distributional Models of Meaning. Computing Meaning, 2014, pp.71–86.

GREFENSTETTE, E. ET AL. New Directions in Vector Space Models of Meaning. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: Tutorials. 2014

GREFENSTETTE, E. Simulating Logical Calculi with Tensors. 2013

HENRIKSSON, A. et al. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. Journal of Biomedical Semantics, 2014, pp.1–25.

HERMANN, K.M. & BLUNSOM, P. The Role of Syntax in Vector Space Models of Compositional Semantics. ACL, 2013, pp.894–904.

HOFMANN, T. Probabilistic Latent Semantic Indexing. Proceedings of the Twenty Second Annual International SIGIR Conference on Resea rch and Development in Information Retrieval, 1999. pp.289–296.

JAUHAR, S.K., DYER, C. & HOVY, E. Ontologically Grounded Multi-sense Representation Learning for Semantic Vector Space Models. NAACL, 2015, pp.683–693.

JURAFSKY, D. & MARTIN, J.H. Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2007

JURGENS, D. & PILEHVAR, M.T. Semantic Similarity Frontiers: From Concepts to Documents. In Conference on Empirical Methods in Natural Language Processing EMNLP. 2015. p. 269.

JURGENS, D. & STEVENS, K. The S-Space Package: An Open Source Package for Word Space Models. Proceedings of the ACL 2010 System Demonstrations, (July), 2010. pp.30–35.

KANERVA, P. Sparse Distributed Memory. MIT Press. 1988

KIELA, D. & CLARK, S. A Systematic Study of Semantic Vector Space Model Parameters. EACL 2014 14th Conference of the European Chapter of the Association for Computational Linguistics Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC). 2014.

KRISHNAMURTHY, J. & Mitchell, T., 2013. Vector Space Semantic Parsing: A Framework for Compositional Vector Space Models.

LEE, D.D., HILL, M. & SEUNG, H.S. Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing Systems. 2000

LIU, N. ET AL. Text Representation: from Vector to Tensor. Proceedings of the Fifth IEEE International Conference on Data Mining, 2005. pp.3–6.

MANNING, C., PRABHAKAR RAGHAVAN & SCHÜTZE, H. An Introduction to Information Retrieval, Cambridge University Press. 2008

MANNING, C.D. Foundations of Statistical Natural Language Processing, Cambridge, Massachusetts: The MIT Press. 1999

MIKOLOV, T., CHEN, K., ET AL. Distributed Representations of Words and Phrases and their Compositionality. NIPS, 2013. pp.1–9.

MIKOLOV, T., CORRADO, G., ET AL. Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR, 2013. pp.1–12.

MIKOLOV, T. Learning Representations of Text using Neural Networks. NIPS Deep Learning Workshop. 2013. pp.1–31.

MIKOLOV, T. & COM, T.G. Distributed Representations of Sentences and Documents. Proceedings of the 31st International Conference on Machine Learning, 32. 2014

MITCHELL, J. & LAPATA, M. Composition in Distributional Models of Semantics. Cognitive Science, 34, 2010. pp.1388–1429.

PENNINGTON, J., SOCHER, R. & MANNING, C.D. GloVe: Global Vectors for Word Representation. Proceedings of the Empirical Methods in Natural Language Processing (EMNLP). 2014

PURANDARE, A. & PEDERSEN, T. Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces. HLT-NAACL 2004 Work- shop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004). 2004

REISINGER, J. & MOONEY, R.J. Multi-Prototype Vector-Space Models of Word Meaning. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, (June), 2010, pp.109–117.

REN, W. & HAN, K. Sentiment Detection of Web Users Using Probabilistic Latent Semantic Analysis. JOURNAL OF MULTIMEDIA, 9(10), 2014. pp.1194–1200.

ROHDE, D.L.T., GONNERMAN, L.M. & PLAUT, D.C. An improved model of semantic similarity based on lexical co-occurrence. Cognitive Science. 2009

SAHLGREN, M. An Introduction to Random Indexing. Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE., 2004. pp.1–9.

SAHLGREN, M. The Word-Space Model Using distributional analysis to represent syntagmatic and paradigmatic relations between words. 2006

SALTON, G., WONG, A. & YANG, C.S. A Vector Space Model for Automatic Indexing. Information Retrieval and Language Processing, Communications of the ACM., 18(11), 1975. pp.613–620.

SEIJO, F.C., LUNA, J.M.F. & GUADIX, J.F.H. Recuperación de Información. Un enfoque práctico y multidisciplinar RAMA, ed., Madrid, Spain. 2011

SOCHER, R. ET AL. Semantic Compositionality through Recursive Matrix-Vector Spaces. EMNLP, (Mv). 2012

STEYVERS, M. & GRIFFITHS, T. Probabilistic Topic Models. Handbook of latent semantic analysis, 427(7), 2004. pp.424–440.

THATER, S., FURSTENAU, H. & PINKAL, M. Contextualizing Semantic Representations Using Syntactically Enriched Vector Models. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. 2010.

TURNEY, P. Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL. NRC Publications Archive. 2001

TURNEY, P.D. From Frequency to Meaning: Vector Space Models of Semantics. Journal of Artificial Intelligence Research, 37, 2010. pp.141–188.

TURNEY, P.D. Semantic Composition and Decomposition: From Recognition to Generation, 2014

WIDDOWS, D. & COHEN, T. The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics. Fourth IEEE International Conference on Semantic Computing (IEEE ICSC2010). 2010

ZOU, W.Y. et al. Bilingual Word Embeddings for Phrase-Based Machine Translation. EMNLP, 2012, pp.1393–1398.