MONOGRAFÍA[i]

“EVALUACIÓN DE LOS MÉTODOS DE MEDICIÒN Y COMPARACIÒN DE VECTORES PARA LA CREACIÒN DE UNA NUEVA FUNCIÒN PARA LA BÙSQUEDA DE INFORMACIÒN DIGITAL”

MATEMATICAS

NO. DE PALABRAS: 3757

CONVOCATORIA: MAYO 2019

[i] Por la presente declaro que soy el autor de este trabajo y que solo he recibido el apoyo permitido por parte de del supervisor asignado por el Coordinador de mi colegio, según lo establece el Bachillerato Internacional. He citado debidamente las palabras, ideas o gráficos que no son mías, se hayan expresado éstas de forma escrita, oral o visual. De no ser así, que mi colegio, el IB o mi conciencia me lo reclamen.

TABLA DE CONTENIDOS

Breve Introducción

El modelo GloVe, publicado en 2014 por un equipo conformado por Jeffrey Pennington, Richard Socher y Christopher Manning, del *Computer Science Departmen*t de la *Universidad de Stanford*, de lo presentado originalmente en la *Conference on Empirical Methods in Natural Language Processing*, provee un acercamiento más enfocado y ordenado a la representación de palabras como vectores en espacios de alta dimensionalidad, como un complemento al modelo Word2Vec, desarrollado el año anterior por un equipo de Google, basado en una red neuronal artificial.

El objetivo del modelo es generar vectores que codifiquen el significado de las palabras presentes en un texto sobre el que se calculan las probabilidades de ocurrencia de una palabra dada otra. Conocidos como *Encajes de Palabras* (del Inglés, *Word Embeddings*), ya que son reducciones arbitrarias de un espacio abstracto con dimensiones iguales al número de palabras en el vocabulario.

La innovación principal del modelo fue su acercamiento más sistemático al problema, pues pese a ser matemáticamente muy parecido al de Google, en la práctica, la generación de los encajes de palabras se realiza de manera que sea entendible para humanos porque el proceso para inferir los significados, no solo los vectores que resultan de este, tienen correspondencia directa con conceptos en la sintáctica y semántica del lenguaje natural, de forma que, con tal de complementar a este objetivo, la investigación presente surge de la pregunta ¿De qué manera la búsqueda en lenguaje natural de información se puede expresar como una relación entre el producto punto y magnitud de vectores de alta dimensionalidad que codifican el significado de las palabras por métodos estadísticos?

2. Funcionamiento y Componentes del Modelo GloVe

1.1 Explicación del Modelo

Según Pennington *et al* (2014) el algoritmo de GloVe aprovecha la estadística global al calcular una matriz de co-ocurrencia palabra-palabra para todo termino en los textos usados en el entrenamiento, recopilando todos los pares de palabras que se repiten en él, además de tomar en cuenta los conteos locales para cada palabra (inspirado en Word2Vec), pues considera un número limitado de palabras circundantes (lo que es análogo al contexto semántico) y que se utilizan para llenar los elementos de la matriz (Ver Figura 1).

Por lo anterior, el primer paso es obtener las co-ocurrencia de cada palabra, para lo cual, con los propósitos ilustrativos, tomaremos por ejemplo, una adaptación de un famoso pangrama en inglés, *"El veloz zorro salta el pozo grande"*, con una ventana de 5 palabras, como se ilustra a continuación.

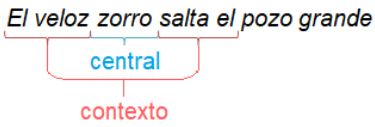

Figura 1: Ejemplificación del conteo con una ventana de 5 palabras

	el	veloz	zorro	salta	pozo	grande
el	2	1	2	1	1	1
veloz	1	1	1	1	0	0
zorro	2	1	1	1	0	0
salta	1	1	1	1	1	0
pozo	1	0	0	1	1	1
grande	1	0	0	0	1	1

Figura 2: Matriz de co-ocurrencia para nuestro enunciado simple

El papel original menciona luego, que la mejor forma de conocer sobre la naturaleza de una palabra, partiendo de la suposición de que ella está directamente relacionada a su contexto, es mediante la probabilidad de que una palabra de prueba, aparezca cerca.

(1)
$$P(j|i) = \frac{X_{ij}}{X_i}$$
$$\mathrm{X_{ij}} = \text{numero de veces que } j \text{ aparece en } i$$
$$X_i = \text{numero de palabras que aparecen con } i$$

Construyendo sobre esto, el algoritmo presentado por Pennington *et al*, adopta, acorde a sus descubrimientos, las razones entre las probabilidades de que estas dos palabras aparezcan en el contexto de una tercera *k*, como la forma de descubrir la naturaleza de una palabra.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Figura 3: Resultados al usar las razones sobre las probabilidades por si solas (Pennington *et al*, 2014)

Consecuentemente las palabras relacionadas a *i* tendrán valores mayores, en cambio, palabras relacionadas a *j* tomaran valores menores, y finalmente, para los que sean neutros, por ser cercanas a ambas o ninguna, aproximaran a uno. De forma que, podemos resumir el modelo como:

(2)
$$F(w_i, w_j, \bar{w}_k) = \frac{P_{ik}}{P_{jk}}$$
$$\text{Donde } w_* \text{ son vectores extraidos de la matriz de co} - \text{ocurrencia}$$

Siendo *F(w)*, la función mediante la cual aproximemos los mejores vectores, con tal que se acerquen a la razón de probabilidad que se ilustra, la forma óptima de inducir que abarque las sub-estructuras lineales que relacionen los significados, permitiendo las analogías que se exponen en la siguiente sección, es mediante una relación algebraica, como se hace en la siguiente ecuación:

(3) $$F\left(w_i - w_j, \bar{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

Destacando que los argumentos de la función son vectores, mientras que se espera que el resultado sea un escalar, Pennington *et al* proponen la introducción del producto punto.

(4) $$F\left((w_i - w_j) \cdot \bar{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

Posteriormente, tratan que la función sea inmune al intercambio de que palabra cumple cualquier rol, es decir que podamos intercambiar entre *i*, *j* y *k* sin problema, de forma que la siguiente ecuación debe ser igual a la previa:

(5) $$\begin{aligned} & F\left((w_i - w_j) \cdot \bar{w}_k\right) = \frac{F(w_i \cdot \bar{w}_k)}{F\left(w_j \cdot \bar{w}_k\right)} \\ & \Rightarrow F(w_i \cdot \bar{w}_k) = P_{ik} \; (Ver\ Eq. 4) \\ & \Rightarrow F(w_i \cdot \bar{w}_k) = P_{ik} = \frac{X_{ik}}{X_i} \; (Ver\ Eq. 1) \end{aligned}$$

De la anterior, notamos que la función apropiada para ser *F(w)*, es, por ser la única que cumple la regla, la función exponencial natural, de modo que la *Eq. 5.1* queda como:

(6) $$e^{(w_i - w_j)\cdot \bar{w}_k} = \frac{e^{w_i \cdot \bar{w}_k}}{e^{w_j \cdot \bar{w}_k}}$$

Y de ahí que *Eq. 5.3 s*e vuelva:

(7) $$e^{w_i \cdot \bar{w}_k} = P_{ik} = \frac{X_{ik}}{X_i}$$

Entonces, simplificamos, para regresar los vectores a primer plano, de modo que:

(8) $$\begin{aligned} & w_i \cdot \bar{w}_k = \log P_{ik} \\ & \Rightarrow\ w_i \cdot \bar{w}_k = \log P_{ik} = \log X_{ik} - \log X_i \end{aligned}$$

En el documento original, los investigadores buscando simetría, añaden un término de sesgo para *k*, en el cual introducen el último logaritmo, por ser independiente de *k*, dejándonos con:

(9) $$w_i \cdot \bar{w}_k + b_i + \bar{b}_k = \log X_{ik}$$

Además, considerando el caso para en el cual el parámetro para el logaritmo sea 0, antes de construir el modelo, proponen la siguiente función de ponderación, por motivos puramente empíricos, para las frecuencias de la matriz de co-ocurrencia.

(10) $$f(x) = \begin{cases} (x/100)^{\frac{3}{4}} & x < 100 \\ 1 & en\ cambio \end{cases}$$

Finalmente, todo esto es resumido en el modelo, una función de error cuadrado promedio con la función de ponderación añadida, de modo que la ecuación sobre la que se itera para aproximar los vectores óptimos que se acerquen a la razón de probabilidades seria $J(w)$:

(11) $$\begin{aligned} & w_i \cdot \bar{w}_k + b_i + \bar{b}_j = \log X_{ij} \\ & \Rightarrow w_i \cdot \bar{w}_k + b_i + \bar{b}_j - \log X_{ij} = 0 \\ & \therefore J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i \cdot \bar{w}_k + b_i + \bar{b}_j - \log X_{ij}\right)^2 \end{aligned}$$

Minimizar la función $J(w)$, es decir, encontrar los vectores w_i y $\bar{w}_k$ cuyo producto escalar sumado a sus respectivos términos de sesgo se acerque al logaritmo de la probabilidad de que tales dos palabras aparezcan juntas (ver Fig. 3) produce un par de matrices con vectores referidos como *principal* y "*de contexto"*, que apropiadamente corresponden, respectivamente, a los vectores calculados para cuando una palabra está en el centro de la ventana, y para cuando aparece entre las palabras circundantes, dado esto, el equipo sugiere promediar ambos vectores para obtener mejores resultados en las tareas subsecuentes.

La optimización anterior es relativamente simple, pues implica únicamente calcular la función de error $J(w)$ con las valores aleatorios iniciales, tras lo cual, se determinan los valores de sus derivadas parciales con respecto a w_i, $\bar{w}_k$, b_i y $\bar{b}_j$ que se introducen en una función de actualización — AdaGrad (Ec. 12), según la investigación — para determinar cómo se modificaran los términos anteriormente referidos.

$$\theta_{t+1} = \theta_t - \frac{\eta \frac{\partial J(t)}{\partial \theta}}{\sqrt{\sum_{\tau=1}^{t} (\frac{\partial J(\tau)}{\partial \theta})^2}}$$

(12)

$$\theta = \text{cualquiera de los 6erminus } w_i, \overline{w}_k, b_i, \text{ ó } \bar{b}_j$$

$$t = \text{punto actual}$$

$$\eta = taza\ de\ aprendizaje$$

Por último, solo se computa el nuevo valor del termino en cuestión (θ_{t+1}) con el algoritmo AdaGrad, que depende del valor actual (θ_t) y la derivada parcial de la función de error $J(w)$ con respecto al término, en el punto t ($\frac{\partial J(\tau)}{\partial \theta}$). La modificación esta modulada por una constante definida de forma arbitraria, según sugiera la experimentación, conocida *taza de aprendizaje* (del inglés, *learning rate).*

1.2 Encajes de Palabras

Como se explicó en la sección anterior, el algoritmo produce una matriz de forma $V \times D$ (ver Figura 1), donde V es el número de todas las diferentes palabras presentes en los documentos del entrenamiento (o vocabulario), y D es el número de propiedades (maldad, bondad, y otros como masculinidad) de las palabras, representadas por números decimales en sus columnas (o dimensiones del espacio en que se describe el vector). En base a lo anterior, y para ajustarse a las limitaciones del papel, la investigación presente se ocupa de una matriz de forma 22×50, un extracto de la original publicada por el equipo de la Universidad de Stanford, que define 400,000 palabras

Figura 4: Ilustración de la matriz del modelo (Adaptado de Alammar, J.)

En cada fila de esta matriz se describen vectores que existirían en un espacio de 50 dimensiones, el cual pese a ser imposible de imaginar, puede construirse por analogía a un espacio bidimensional o tridimensional, similar al expuesto en la Figura 2. Estos vectores-fila pueden ser trazados como vectores de posición, desde el origen, volviendo posible realizar operaciones matemáticas como sumas y restas que, de forma abstracta, sean correspondientes a alguna relación significativa entre palabras.

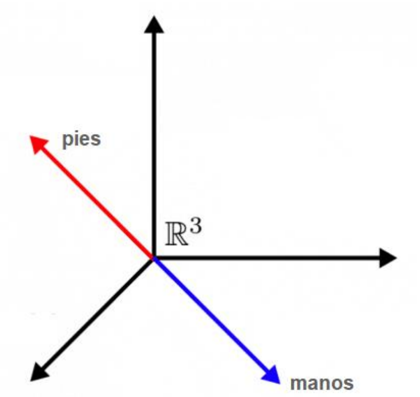

Figura 5: Ejemplo de vectores-palabra en un modelo descrito en base a $\mathbb{R}^3$**.** (Adaptado de Alammar, J.)

Teniendo en mente el funcionamiento interno del modelo descrito en la sección previa, se recordará que las palabras que aparecen en similar contextos, o lo que es lo mismo, en esta perspectiva más sistemática para con la sintaxis, en esencia, la intersección de los conjuntos de palabras circundantes de acuerdo a una ventana de contexto para dos palabras centrales $w0, w1$, siendo este mayor a un umbral antes establecido. Esto también implica que los valores para las filas del mismo par de palabras en la matriz de co-ocurrencia son numéricamente cercanos.

(13) $$|w0_{contexto} \cap w1_{contexto}| > umbral$$

Así, dos palabras cuyo que cumplen la regla, es decir, cuyos patrones en la matriz de co-ocurrencia son semejantes resultan, en vectores numéricamente similares, a continuación:

libro	-0.01	0.93	-0.73	-0.55	0.77	0.36	-1.14	-1.16	0.34	0.29	-0.87	0.92	-0.47	-0.23	1.48	-0.82	-0.17	-0.51	-0.28	0.23	0.72	0.23	0.49	0.35	0.77
historia	0.48	0.88	-0.23	0.03	0.80	0.43	-0.61	-0.61	-0.43	-0.01	-1.29	0.53	-0.83	0.31	1.20	-0.48	-0.47	-0.20	-0.28	0.35	0.46	0.77	0.01	0.55	1.00

libro	-1.44	-1.75	-0.29	-0.10	-0.37	2.55	-1.06	-0.05	-0.26	-0.63	0.03	-0.19	0.20	-0.26	-0.41	0.83	-0.14	-0.28	0.10	-0.17	0.52	0.32	-0.09	-0.27	-0.01
historia	-1.40	-1.69	0.30	0.61	-0.46	2.60	-1.22	0.29	-0.46	-0.26	0.38	-0.28	-0.48	-0.06	-0.59	0.26	0.21	-0.02	-0.30	-0.19	0.54	0.75	-0.41	0.24	0.26

Figura 6: Comparación lado a lado de dos vectores para palabras similares con escala de colores

1.3 Set de Datos SimLex-999

Como se mencionará, por el momento la métrica predeterminada para realizar una comparación de dos palabras representadas por vectores como los que aquí se tratan, es la función conocida como similitud coseno (ver Eq. 15).

Aunque la fundamentación teórica (con analogías geométricas) para justificar el funcionamiento de la función es correcta, en la práctica notamos que los resultados que da la función se alejan considerablemente de los valores que asignaría un humano al mismo par de palabras.

Lo anterior se debe a que, como se ha mostrado, los vectores del modelo se crean en un cuerpo de texto en el que el entendimiento de *contexto*, se limita a las palabras circundantes. Estas palabras circundantes, son las que establecen el significado, por lo que, dado que antónimos y sinónimos aparecen rodeados de las mismas palabras, el algoritmo les asignara vectores cercanos, por tanto el ángulo (y su coseno) será pequeño, pese a que nosotros asignaríamos un valor diametralmente opuesto para cada uno, y de ahí, una similitud casi nula.

En respuesta a este problema, un equipo de la universidad de Cambridge desarrollo el set de datos *SimLex-999*, compuesto, como sugiere su nombre, de 999 pares de palabras de diferente categoría: 666 sustantivos, 333 verbos y 111 adjetivos.

A diferencia de la información generada sin verificar que provee la función de la similitud coseno sobre un par de palabras, la similitud que se indica para cada par en este conjunto de datos es el resultado de la recopilación de la opinión de 500 sujetos humanos a través de la plataforma Amazon Mechanical Turk.

2. Relaciones e Implicaciones de los Vectores y el Modelo

2.1 Analogías y Similitud

La repercusión directa de lo discutido en el capítulo anterior, es que el modelo tiene la capacidad de componer abstracciones semánticas de bajo nivel, o simples analogías bidireccionales que pueden ser formuladas como *"$w1$ es a $w2$, como $w3$ a $w4$"*.

Por ejemplo, siguiendo la misma estructura, considérese la oración *"dedos es a mano, como $w3$ es a pies"*[1], que puede descomponerse a una operación matemática que en términos del espacio vectorial conserva las propiedades de aquella formulada en lenguaje natural, como:

(14) $$\boldsymbol{dedos} - \boldsymbol{mano} + \boldsymbol{pies} = \boldsymbol{dedos\_piesE} \approx w3\ (\boldsymbol{dedos\_pies})$$

En cuanto al modelo, el enunciado anterior está justificado por el hecho de que el vector producido para una palabra dada (se asume) engloba todas sus características en relación a las demás del vocabulario, de esa forma, la analogía puede considerarse como: la parte de la definición de $w1$ que no es común a $w2$, con la adición del concepto descrito por el vector $w4$.

En este caso, la parte de la definición de *dedos* que no intersecta con la de *mano* es conservada, como se expone debajo:

[1] Del original en inglés, donde *dedos_pies* es escrito como *toes*, y la analogía completa se simplifica en "*fingers* is to *hand*, as *[toes]* is to *feet*".

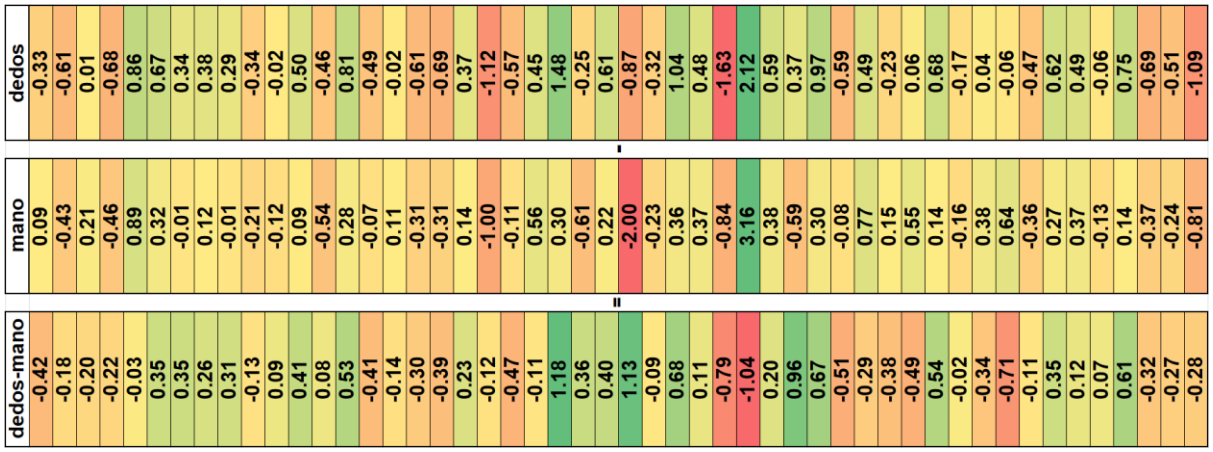

Figura 7: Resta de vectores

Subsecuentemente, las características de *pies* son sumadas al vector resultante:

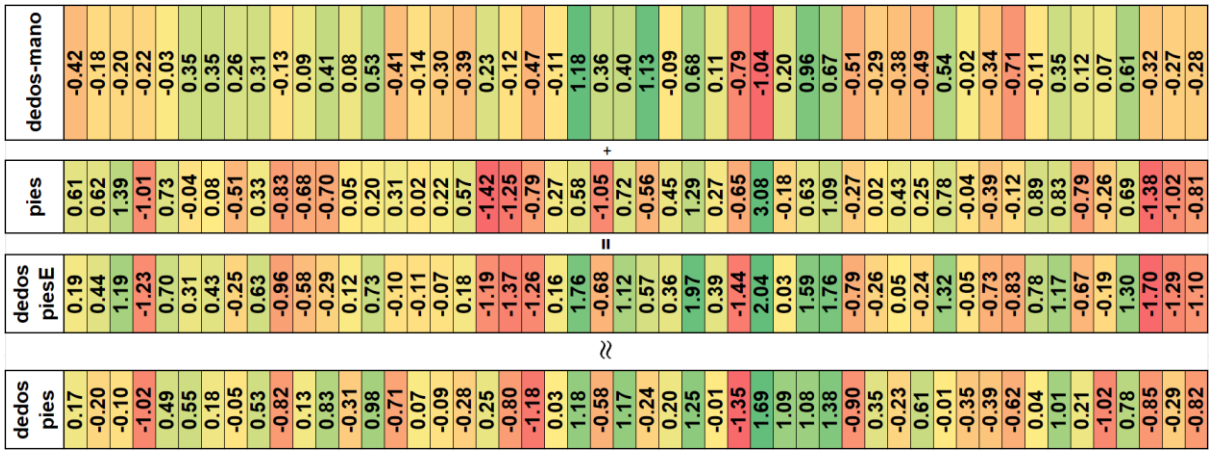

Figura 8: Suma de vectores, cuyo resultante estimado es comparado con el vector real para *toes*

Como se puede percibir a través de la escala de colores, los vectores **dedos_pies** y **dedos_piesE**, son comparables en cuanto a sus elementos, como conjuntos, pero yendo más allá, viéndolos en el marco del modelo, podemos calcular el ángulo entre estos dos vectores, operación ahora estándar en el procesamiento de lenguaje natural, devoto a estas representaciones, con la formula a continuación.

(15)
$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N, \qquad sim(\boldsymbol{x}, \boldsymbol{y}) = cos(\theta) = \frac{\boldsymbol{x} \cdot y}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = \frac{\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{y}_i}{\sqrt{\sum_{i=1}^{N} \boldsymbol{x}_i{}^2 \sum_{i=1}^{N} \boldsymbol{y}_i{}^2}}$$

(16) $$\sum_{i=1}^{50}(\boldsymbol{x}_i)(\boldsymbol{y}_i) = (0.17)(0.19) + \cdots + (-0.82)(-1.10) \\ = 0.03 + \cdots + 0.9 = 27.45$$

(17) $$\sum_{\boldsymbol{i=1}}^{\boldsymbol{50}} \boldsymbol{x_i}^{\boldsymbol{2}} = 0.17^2 + \cdots + -0.82^2 \\ = 0.03 + \cdots + 0.67 = 26.99$$

(18) $$\sum_{\boldsymbol{i=1}}^{\boldsymbol{50}} \boldsymbol{y_i}^{\boldsymbol{2}} = 0.19^2 + \cdots + -1.10^2 \\ = 0.03 + \cdots + 1.20 = 46.02$$

(19) $$sim(\boldsymbol{dedos\_pies}, \boldsymbol{dedos\_piesE}) = \frac{27.45}{\sqrt{(26.99)(46.02)}} = \frac{27.45}{\sqrt{1242.01}} \approx 0.78$$

Respecto a la imagen del coseno $sim^{\rightarrow}: \mathbb{R} \rightarrow [-1,1]$, tenemos que mientras más lejano sea el resultado de 0, la similitud es mayor, y lo opuesto es también verdadero, de forma que podemos decir que una correlación de ~78% supera el umbral estándar de 70%, y por lo tanto, efectivamente, los vectores siendo semejantes, prueban que el modelo puede describir relaciones semánticas a través de operaciones aritméticas sencillas.

2.2 Apología para la Sección Previa

Ahora bien, teniendo clara la analogía geométrica, el concepto de que la similitud se puede medir con el coseno del ángulo entre los objetos parece razonable, pero ¿cuál es la justificación para esto? Quizá la respuesta se nos presente más clara con una comparación con la fórmula para la correlación de Pearson, en cuanto a lo simbólico, dada por:

(20) $$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N, \qquad r(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{N}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{y}_i - \overline{\boldsymbol{y}})}{\sqrt{\sum_{i=1}^{N}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^2 \sum_{i=1}^{N}(\boldsymbol{y}_i - \overline{\boldsymbol{y}})^2}}$$

Como puede observarse, ambas (Ec. 14 y 19) son muy similares simbólicamente, incluso son idénticas cuando los vectores que se introducen están normalizados, o lo que es lo mismo, cuando la media es 0, y pese a esto, fueron concebidas con propósitos distintos. Siguiendo del punto de la media, la sustracción de esta en $r(\boldsymbol{x}, \boldsymbol{y})$ significa que solo se mide la similar variación entre los datos, y no una similitud numérica absoluta, que es lo que hace más adecuada la similitud coseno.

La comparación cobra más sentido al graficarse, pues nos deja ver que podemos tomar los componentes de los vectores y marcarlos en un plano bidimensional, de forma que el problema se vuelve una optimización para encontrar la mejor línea recta, dando como resultado un coeficiente de Pearson casi idéntico a la similitud coseno.

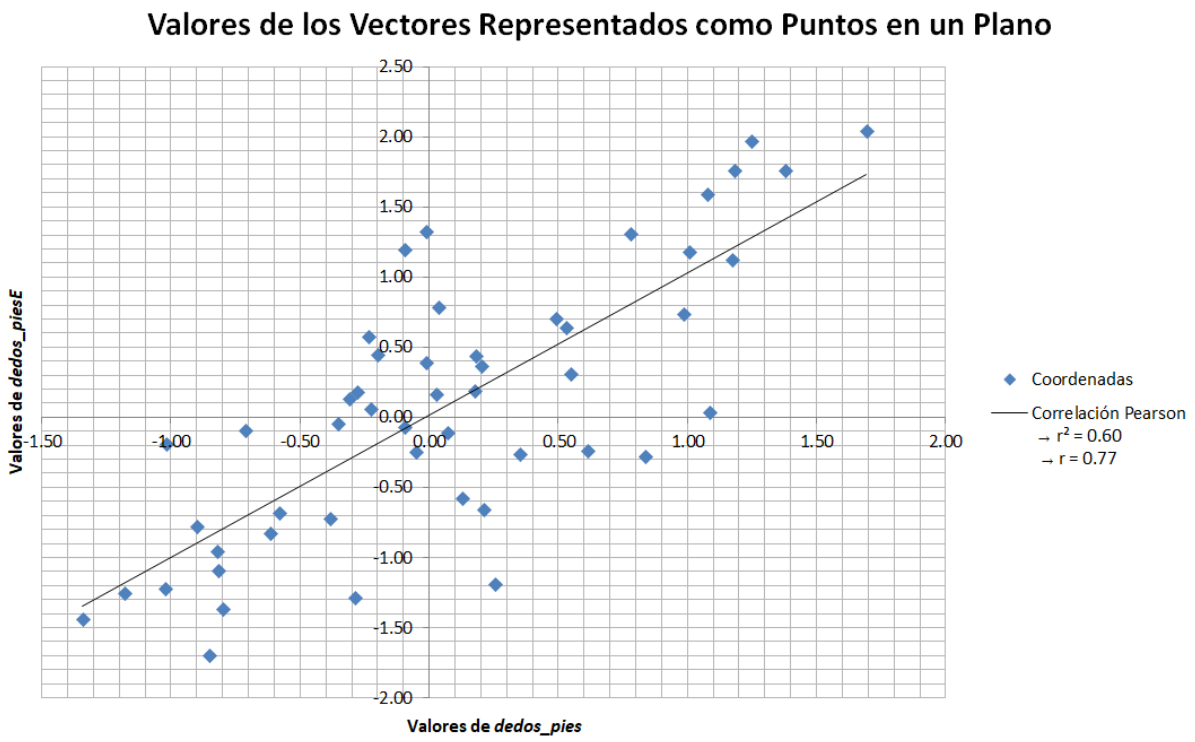

Figura 9: Similitud de los vectores (con sus componentes como puntos) representada en una correlación

De manera que la similitud coseno efectivamente se presenta como la manera más apropiada de comparar los vectores del modelo, a través de una compleja abstracción para el concepto de los ángulos en alta dimensionalidad, pero que puede explicarse igualmente en otros términos más tangibles.

2.3 Explorando Diferentes Métricas

Al realizar una comparación, la mejor forma disponible es encontrar los puntos de comparación mediante preguntas estandarizadas, es decir que se pueden cuantificar para cada objeto igualmente y sin distinción *¿cuánto tiene de x?* o *¿qué tan y es?* De forma que se identifiquen en cada uno las características principales por las que guiar el veredicto. Similarmente, en el caso presentado, como ya se mostró, se puede preguntar *¿cuál es el ángulo entre los vectores?* inclusive, ir más allá, como *¿cuánto mide cada vector?* La pregunta importante aquí, es hasta qué punto preguntar esto es desaconsejable, y como puede tornarse útil.

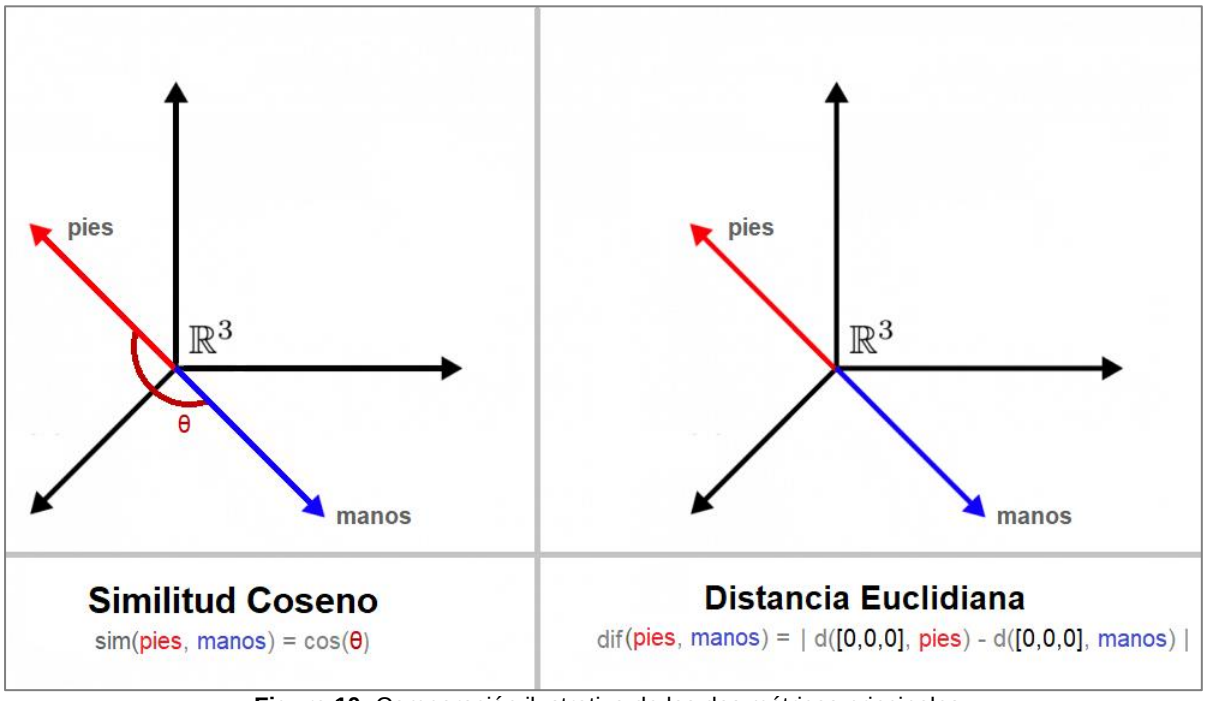

Figura 10: Comparación ilustrativa de las dos métricas principales

Sabemos que un vector describe una magnitud con cierta dirección, independientemente de la posición, por lo que podemos medir la extensión que tiene este, incluso sin tener los puntos donde comienza y termina, usando la definición estándar de la *distancia Euclidiana*:

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N, \qquad d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x}) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_N - y_N)^2}$$

$$\Rightarrow dif_{euclid}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{z}) = |d(\boldsymbol{x}, \boldsymbol{y}) - d(\boldsymbol{w}, \boldsymbol{z})| \tag{21}$$

Utilizando la formula anterior, podemos calcular el desplazamiento que describen los vectores $\boldsymbol{x}$ y $\boldsymbol{y}$, es decir, conocer cuántas unidades se extiende cada uno, y de ahí encontrar el valor absoluto de la diferencia de sus longitudes, de forma que mientras menor sea el valor (Ec. 21.2), mayor será la similitud. Ahora bien, el problema aparece rápidamente, utilizar una métrica significa sacrificar el conocimiento de la otra propiedad del vector, esto es, que al usar la similitud coseno, distinguimos la dirección de cada uno, más no la longitud, y lo contrario es cierto para la distancia Euclidiana.

Entonces ¿qué es lo que hace que la comunidad se incline por la primera métrica, y rotundamente rechace la segunda? Podríamos argumentar que es simplemente por cuestiones de elegancia, pues es preferible comparar la semejanza que la disimilitud, pero no sería la historia completa. La respuesta, es la misma naturaleza del modelo, porque si graficamos las palabras del cuerpo de texto sobre el que se entrenó el modelo originalmente (ver Fig. 10), notamos una tendencia de que la magnitud de los vectores aumenta de manera directamente proporcional a la frecuencia con que aparecen las palabras que representan, de forma que, pareciese que conocer esta propiedad del vector no es muy útil para la comparación de significados, ya que en realidad, sirve más como una medida de estadística sobre los datos en que se desarrolló el modelo.

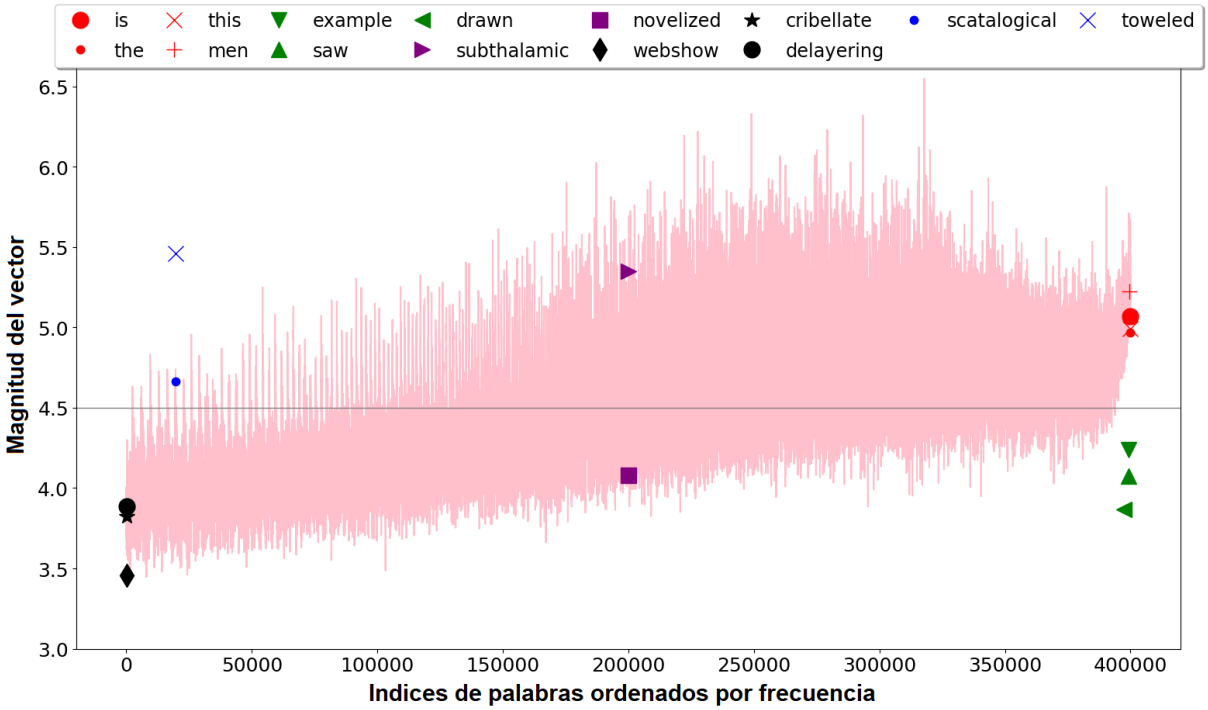

Figura 11: Frecuencia de las palabras en el texto original contra la magnitud de los vectores producidos (con palabras altamente frecuentes como artículos a la derecha extrema, palabras complejas a mitad, y otras altamente improbables a la extrema izquierda).

La imprecisión de la distancia como métrica se vuelve incluso más clara con la gráfica anterior, pues podemos apreciar que palabras como *is* y *men* (del inglés para *es* y *hombres*, respectivamente) se tomarían por virtualmente idénticas — ya que sus marcadores se superponen, indicando que tiene casi igual magnitud — de manera que no se puede emplear independientemente, pues conduciría a errores y contradicciones.

2.4 Discriminación de Resultados para Justificar la Distancia

En la sección anterior exploramos la distancia como una forma de comparar vectores, y al encontrar que esta se veía fuertemente influida por la frecuencia de aparición de las palabras en el texto sobre el que se "entrenó" el modelo, se descartó, por introducir ruido. Algo que se pasa por alto en tal conclusión, es el punto que se mencionaba tan solo unos párrafos antes, de que usar una forma de medición significa sacrificar el conocimiento de la otra propiedad, por lo que, descartando la distancia, usando solo el coseno del ángulo entre los vectores como medida, no hay nada que ajuste para la frecuencia, salvo por la función de ponderación dentro de la función de error del modelo (Ec. 10).

A razón de esto, sirva esta investigación para introducir una nueva noción de la distancia entre dos vectores que representan palabras (en el contexto de GloVe),

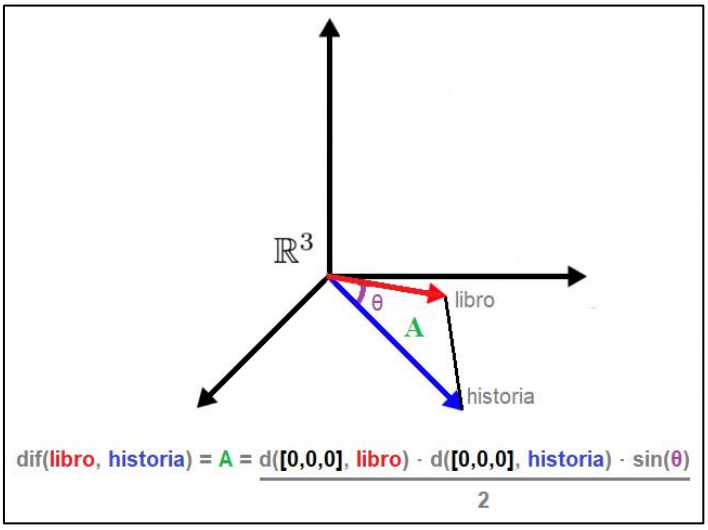

Figura 12: Primeras nociones de la nueva función

Para crear la función anterior, se combinan las ideas anteriores, principalmente de emplear la distancia euclidiana – pues como muestra la Fig. 11, provee información importante para discriminar resultados por frecuencia – y el ángulo que se forma entre los vectores que se tratan. De manera que, por conveniencia, se toma

inspiración en una noción geométrica, es decir, la fórmula para calcular el área de un triángulo a partir de un ángulo y dos lados, acoplada para la situación en mano.

Ahora bien, la continua experimentación con esta fórmula sobre el set *SimLex-999* arroja un error absoluto superior a la similitud coseno a solas. Entonces, a razón de tales hallazgos, se reemplaza el uso de la función $\sin\theta$ por $\cos\theta$, que pese a desviarse de la inspiración geométrica, se acerca más a la función base (ver Ec. 15).

Dada la naturaleza del concepto tras la función (encontrar un área) es imperativo que el resultado sea siempre positivo, por lo que, partiendo de la Ec. 15 de la similitud coseno, que puede devolver un valor entre $[-1,1]$, será trasladada a poder representar un área, por lo que se introduce una corrección que lo mantenga en un rango positivo. A continuación:

(22) $$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N, \qquad dist(\boldsymbol{x}, \boldsymbol{y}) = 1 - cos(\theta)\ \ (Ver\ Ec. 15)$$

Teniendo esto, arribamos a la versión final de la función, que por lo anterior ha sido modificada para asimilarse a la inicial y permanecer positiva, como se muestra a continuación.

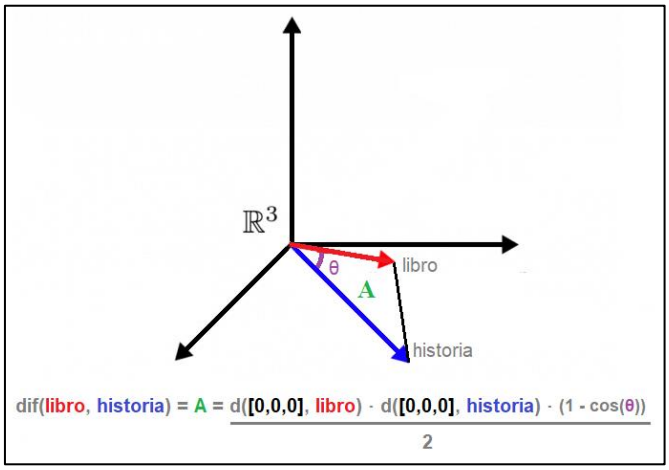

Figura 13: Iteración final de la nueva función provisional $dif(\boldsymbol{x}, \boldsymbol{y})$

Consecuentemente, es necesario considerar que el área del triángulo puede crecer de forma infinita, mientras que para efectos de comparación, como similitud coseno, debe tener un rango propiamente delimitado, por lo que se le realiza la siguiente adecuación, para definirla entre $[0, 1]$.

(23)
$$\frac{1}{1 + dif(\boldsymbol{x}, \boldsymbol{y})}$$

Finalmente, para efectos de experimentación y poder equiparar los resultados con aquellos que otorga la similitud coseno (ver Eq. 15) se transforma de la escala anterior a la de $[-1, 1]$.

(24)
$$\frac{\left(\left(\frac{1}{1 + dif(\boldsymbol{x}, \boldsymbol{y})}\right) - 0\right) * (1 - -1)}{1 - 0} - 1$$
$$\Longrightarrow 2\left(\frac{1}{1 + dif(\boldsymbol{x}, \boldsymbol{y})}\right) - 1$$
$$\Longrightarrow \frac{2}{1 + dif(\boldsymbol{x}, \boldsymbol{y})} - 1$$
$$\therefore diferencia\_triangular(\boldsymbol{x}, \boldsymbol{y}, \theta) = \frac{2}{1 + \frac{d(\boldsymbol{x}) * d(\boldsymbol{y}) * (1 - \cos\theta)}{2}} - 1$$

2.5 Comparando la Nueva Función

Ahora bien, con la iteración final de la función, solo resta realizar las pruebas con el siguiente extracto al azar del set de datos *SimLex-999:*

			Similitud SimLex-999		Triangular		Coseno	
Tipo	Palabra1	Palabra2	Original [0,10]	**Ajustada [-1,1]**	**Diferencia [-1,1]**	Error	**Similitud [-1,1]**	Error
Adjetivo	old	new	1.58	**-0.684**	**-0.6681**	0.016	**0.620**	1.304
Adjetivo	sad	terrible	5.4	**0.08**	**-0.4751**	0.555	**0.772**	0.692
Sustantivo	wife	husband	2.3	**-0.54**	**0.1744**	0.714	**0.951**	1.491
Sustantivo	woman	man	3.33	**-0.334**	**-0.2498**	0.084	**0.886**	1.220
Sustantivo	dog	cat	1.75	**-0.65**	**0.0885**	0.739	**0.922**	1.572
Sustantivo	floor	ceiling	1.73	**-0.654**	**-0.6588**	0.005	**0.675**	1.329
Sustantivo	phrase	word	5.48	**0.096**	**0.0124**	0.084	**0.916**	0.820
Sustantivo	wealth	poverty	1.27	**-0.746**	**-0.734**	0.012	**0.576**	1.322
Sustantivo	sorrow	shame	4.77	**-0.046**	**-0.4547**	0.409	**0.767**	0.813
Sustantivo	guy	girl	3.33	**-0.334**	**0.749**	1.083	**0.522**	0.856
Verbo	verify	justify	4.08	**-0.184**	**-0.642**	0.458	**0.573**	0.757
Verbo	understand	listen	4.68	**-0.064**	**-0.508**	0.444	**0.753**	0.817
PROMEDIO						**0.602**		**0.704**

Figura 14: Comparación del rendimiento de cada función

Para las pruebas se tomaron 12 pares de palabras al azar, que fueran representativos de toda la variedad del set de datos (ya que se desglosa en 666 sustantivos, 222 verbos y 111 adjetivos). Dado que la similitud entre las palabras esta expresada en una escala del 1 al 10, y puesto que la función con la que buscamos comparar – principalmente similitud coseno, ver Eq. 15 – produce un resultado entre [-1, 1], se ajustó la escala a como se muestra en la columna 5° (además de la corrección previamente realizada a la función original de diferencia triangular).

Como puede observarse (Fig. 14) la función desarrollada en este trabajo supera en gran medida a la similitud coseno. La comparación fue realizada usando la siguiente fórmula para encontrar el error absoluto.

(24) $$error(actual, ideal) = |ideal - actual|$$

El error absoluto es menor para todos los pares de palabras, salvo el 10° (*guy – girl*), pero de cualquier manera, relativo al resto y en comparación al resultado de la similitud coseno, no se encuentra muy alejado del valor esperado.

El error promedio, que se indica en la última fila (15°) fue calculado sobre el total de pares que ofrece el set de datos (es decir, 999) para volver más efectiva la comparación. El programa empleado se muestra en los anexos (5.1)

Por lo que, para cerrar, con una disminución del error promedio de aproximadamente 0.102 unidades, la experimentación relatada y el desarrollo de la nueva función (diferencia triangular, ver Eq. 24) puede tomarse por efectiva y superior a la previa similitud coseno.

5. Anexos

5.1 Código del Experimento en Python 3.6

```
import statistics, math, csv

def remap( x, oMin, oMax, nMin, nMax ):
    return (x-oMin)*(nMax-nMin)/(oMax-oMin) + nMin

def cosine_sim(word1, word2):
    a = model[word1]
    b = model[word2]
    d_word1 = np.linalg.norm(a)
    d_word2 = np.linalg.norm(b)
    return np.dot(a, b)/(d_word1*d_word2)

def triangle_distance(word1, word2):
    a = model[word1]
    b = model[word2]
    d_word1 = np.linalg.norm(a)
    d_word2 = np.linalg.norm(b)
    cos_sim = cosine_sim(word1, word2)
    area = (d_word1 * d_word2 * (1 - cos_sim))/2
    return remap((1)/(1 + area), 0, 1, -1, 1)

def error(num1, num2):
    return abs(num1 - num2)

with open("SimLex-999.txt") as f:
    lines = [x for x in csv.reader(f, delimiter="\t", quotechar='"')][1:]
    ## word1, word2, similarity (0, 10)
    full_error_cos = []
    full_error_trig = []
    lx = ["floor", "wealth", "old", "woman", "sorrow", "guy", "understand", "verify", "sad", "phrase", "dog", "wife"]
    ly = ["ceiling", "poverty", "new", "man", "shame", "girl", "listen", "justify", "terrible", "word", "cat", "husband"]

    for x, y, _, z, *rest in lines:
        e = remap(float(z), 0, 10, -1, 1)

        w = triangle_distance(x,y)
        v = cosine_sim(x,y)

        d = error(e, w)
        d2 = error(e, v)

        if x in lx and y in ly:
            print(x + " " + y + " ############## " + str(e) + " ######### " + str(d) + " ###### " + str(d2))

        full_error_trig.append(d)
        full_error_cos.append(d2)

    print("trig_error:" + str(statistics.mean(full_error_trig)))
    print("cos_error:" + str(statistics.mean(full_error_cos)))
```

6. Referencias

5.1 Bibliografía

McCune, B., & Grace, J. B. (2002). Distance Measures. Analysis of ecological communities. Recuperado en Mayo 14, 2019, de https://www.umass.edu/landeco/teaching/multivariate/readings/McCune.and.Grace.2002.chapter6.pdf

Pratap, D. (2017). Content-based Filtering. Statistics for Machine Learning. Recuperado en Mayo 14, 2019, de https://learning.oreilly.com/library/view/statistics-for-machine/9781788295758/eb9cd609-e44a-40a2-9c3a-f16fc4f5289a.xhtml

5.2 Hemerografía

Hill, F., Reichart, R., & Korhonen, A. (2015). SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. Computational Linguistics, 41(4), 665–695. doi: 10.1162/coli_a_00237

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). doi: 10.3115/v1/d14-1162

5.3 Misceláneo

Alammar, J (2018). The Illustrated Word2vec. Recuperado en Mayo 1, 2019 de https://jalammar.github.io/illustrated-word2vec/

Kurita, K. (2018, Abril 29). Paper Dissected: "Glove: Global Vectors for Word Representation" Explained. Recuperado en Mayo 14, 2019, de https://mlexplained.com/2018/04/29/paper-dissected-glove-global-vectors-for-word-representation-explained/.

Prabhakaran, S. (2018, Octubre 30). Cosine Similarity - Understanding the math and how it works? (with python). Recuperado en Mayo 14, 2019, de https://www.machinelearningplus.com/nlp/cosine-similarity/

Tseng, C. (1999, Marzo 15). Sparse Vectors. Recuperado en Mayo 14, 2019, de https://www.cs.umd.edu/Outreach/hsContest99/questions/node3.html