Evaluación Final de PLN
Modelo de Espacio Vectorial

Laura Rodríguez Navas
rodrigueznavas@posgrado.uimp.es

Mayo 2020

Índice

1 Motivación y definición del MEV **2**

1.1 Motivación 2

1.2 Definición del MEV 2

2 Tipos de MEV **3**

2.1 Modelo Booleano 3

2.2 Modelo Probabilístico 5

2.3 Modelo Vectorial 6

3 Medidas de cálculo de relevancia **9**

4 Cálculo de similitud entre vectores de un MEV **9**

4.1 Mediante el producto escalar 10

4.1.1 Modalidad de pesos binarios 10

4.1.2 Modalidad de pesos TF-IDF 10

4.2 Mediante la fórmula del coseno 11

4.3 Mediante el coeficiente de Dice 12

4.4 Mediante el coeficiente de Jaccard 12

5 Espacios vectoriales basados en *word embeddings* **13**

6 Algunas de las aplicaciones de los *word embeddings* **14**

6.1 Word2vec 14

6.2 Doc2vec 14

6.3 FastText 14

6.4 GloVe 15

7 Reflexión personal sobre la utilidad y futuro de los MEV **15**

1 Motivación y definición del MEV

1.1 Motivación

El uso del MEV para el análisis semántico y la representación de la semántica o significado de una palabra, oración o documento es fundamental tanto en el ámbito investigador como para el desarrollo de tareas de PLN industriales de calidad, por lo que el conocimiento de su base teórica, de sus aplicaciones actuales, y de su uso en escenarios futuros es muy importante para cualquier Ingeniero en Informática y, en especial, para aquellos que se quieran especializar en Inteligencia Artificial.

El MEV ha sido ampliamente utilizado en Recuperación de Información, en Categorización de Textos e incluso en Desambiguación, y todavía no ha desarrollado todo su potencial en PLN.

Las preguntas formuladas a continuación nos pueden dar una aproximación de la importancia que tienen actualmente el uso del MEV. Por ejemplo,

- ¿Cómo recupera un buscador como Google o Yahoo! documentos relevantes a partir de una consulta enviada?
- ¿Cómo puede procesar una empresa los reclamos que le dejan sus usuarios en sus portales Web?
- ¿Cómo podemos agrupar los comentarios emitidos en un foro y analizar las opiniones de la gente?

En la actualidad, se está elaborando un prolífico y prometedor trabajo de investigación, que va desde la investigación fundamental con el desarrollo teórico de la semántica distribuida, a las aplicaciones en diversas tareas de clasificación textual mediante el uso de representaciones continúas de palabras o vectores densos de palabras (*word embeddings*).

Es de gran importancia este trabajo de investigación, por ejemplo, una colección de documentos (p. ej. páginas web) aún no está preparada para realizar directamente operaciones de Recuperación de Información. Se necesitan crear nuevas estructuras persistentes que permitan acceder eficientemente a los datos ya procesados.

1.2 Definición del MEV

En un modelo de espacio vectorial (MEV), los documentos se almacenan como vectores de términos y pueden encontrarse en un espacio vectorial de n dimensiones, en grupos, formando colecciones de documentos de acuerdo con la relevancia para una misma clase de necesidad de información. Es decir, cuando queremos acceder a cierta información usando una consulta, esa consulta llega a ser comparada con cada elemento de la colección, y si se obtiene un alto grado de coincidencia, como consecuencia el documento tendrá más probabilidades de ser relevante, y se nos devolverá lo que buscamos.

El proceso para construir un MEV comienza con la extracción de los términos de indización (los términos que van a ser utilizados para describir el contenido del documento). La extracción más simple consiste en considerar todas las palabras aisladas que aparecen en el texto como los términos de indización. Habitualmente se eliminan algunas palabras, las denominadas palabras vacías, entre las que suelen figurar números, preposiciones, conjunciones, etc.

Una vez extraídas esas palabras, se ordenan por orden alfabético y se guardan en una matriz, junto con la referencia del documento de donde proceden (normalmente un número de documento asignado previamente por el modelo). Si se repite este proceso con todos los documentos de la colección, se obtiene finalmente la matriz que almacena los siguientes datos:

- En primer lugar, los términos de indización que aparecen en toda la colección ya sean los propios textos, los resúmenes de los textos del fondo y/o los títulos.
- En segundo lugar, cada uno de estos términos incorpora una lista con los números de los documentos en los que aparece.

Un ejemplo. El objetivo principal de una colección de n documentos indexados por m términos que se pueden representar por una matriz A de dimensión n x m, donde cada elemento a_{ij}, se define por una frecuencia ponderada del término i en el documento j para mejorar el rendimiento en la habilidad de recuperar información relevante y descartar información irrelevante.

La siguiente tabla muestra la matriz A de ejemplo, donde cada fila representa un término en la colección, cada columna un documento y cada celda o elemento de la matriz la ocurrencia del término en el documento.

	D1	D2	D3
T1	1	0	0
T2	0	0	1
T3	1	1	1

Tabla 1: Matriz A.

En ella podemos ver que el término T1 aparece en el documento D1, pero no en los documentos D2 y D3, y que cada fila de la matriz de 3×3 puede ser representada en un espacio de tres dimensiones donde cada elemento e_{ij} de la matriz queda definido como:

$$e_{ij} = l_{ij} * g_i * d_j^-1$$

l_{ij} es el peso local del término i en el documento j, que mide la importancia de dicho término en el documento. g_i es el peso global del término i en la colección de documentos y d_j es el factor de normalización para el j-ésimo documento.

2 Tipos de MEV

2.1 Modelo Booleano

En la tabla de ejemplo (ver Tabla 1) de la sección anterior, se muestra un tipo de MEV: el modelo booleano. Este modelo constituye el primer modelo teórico de RI, el más antiguo, empleado para establecer un subconjunto de documentos relevantes. Al mismo tiempo es, sin duda, uno de los modelos más sencillos tanto desde un punto de vista teórico como práctico, por una parte, al basarse en la teoría de conjuntos y en el álgebra de Boole, y por otra parte al ser fácil de diseñar e implementar.

Una propiedad importante del modelo booleano es que no puede efectuar ningún proceso de ordenación con los documentos resultantes de las consultas. Esta característica suele denominarse **equiparación exacta**, impidiendo que se pueda situar en primer lugar aquel documento posiblemente más útil o relevante para el usuario y relegando a las últimas posiciones a aquellos otros documentos con menos probabilidades de ser relevantes en relación a las consultas.

También se denomina modelo binario porqué considera exclusivamente la presencia (con el número 1) y la ausencia (con el número 0) de los términos en los documentos. Pero eso es el gran responsable de la equiparación exacta, siendo considerada la principal desventaja del modelo.

A pesar de esta desventaja, hoy en día sigue constituyendo el modelo más utilizado. Muchos motores de búsqueda en la web se basan en este modelo, por ser de desarrollo sencillo (como hemos visto, en su versión básica solamente involucra el empleo de una matriz y una interfaz de consulta que permita realizar consultas expresadas mediante palabras o expresiones booleanas), fácil de utilizar por parte de un usuario medio (basta introducir palabras relativas a la necesidad informativa), y bastante eficaz en los resultados obtenidos (en gran parte debido al volumen ingente de documentación presente en la red, lo que provoca la reducción de la respuesta a los documentos que satisfagan estrictamente las condiciones de la consulta.

Volviendo al ejemplo de la sección anterior (ver Tabla 1) vemos que de la consulta de la matriz A únicamente se puede saber si un determinado término de indización está presente (en cuyo caso se simboliza con el número 1) o no lo está (en cuyo caso se simboliza con el número 0) en cada uno de los documentos de la colección, cuyos datos básicos son:

	D1	D2	...	D_n
T1	1	0	...	1
T2	0	1	...	0
...	...	...	...	...
T_n	0	1	...	1

Tabla 2: Matriz A.

donde T1, T2, ..., T_n son los términos de indización empleados en la colección de documentos D1, D2, ..., D_n y donde el número 1 significa que el término correspondiente aparece en ese documento concreto, mientras que el número 0 significa que el término no aparece en dicho documento. En este caso, no se tiene en cuenta la frecuencia de aparición de los términos en los documentos. Se entiende entonces porqué se denomina modelo binario, pues únicamente se juega con dos posibilidades: la aparición y la no aparición de los términos en los documentos.

Si observamos la tabla anterior (ver Tabla 2), podemos deducir de ella las dos representaciones empleadas al manejar un modelo binario. Por una parte, cada término de indización se representa por la lista de documentos en los que aparece, lo que implica la observación de la tabla por filas:

$$\begin{aligned} T1 &= \{D1, ..., D_n\} \\ T2 &= \{D2, ..., D_n\} \\ &\vdots \\ T_n &= \{D2, ..., D_n\} \end{aligned}$$

Por otra parte, cada documento se representa por una lista de ceros y unos, además de sus correspondientes términos de indización que contiene, lo que implica la observación de la tabla por columnas:

$$\begin{aligned} \text{D1} &= \{1, 0, ..., 0\} \\ \text{D2} &= \{0, 1, ..., 1\} \\ &\vdots \\ \text{D}_n &= \{1, 0, ..., 1\} \end{aligned}$$

Se comprende bien ahora porqué se dice que en el modelo binario todo documento se representa mediante una serie ordenada de ceros y unos, tantos como términos se empleen en una colección. Des del primer número que siempre corresponderá a T1, el segundo número que corresponderá a T2, y así sucesivamente hasta llegar a T_n, siendo n el número de términos distintos que representan el contenido de esa colección.

2.2 Modelo Probabilístico

El modelo probabilístico fue introducido en la década de los setenta por Robertson[1] y Sparck Jones[2]. También es conocido como modelo de recuperación de independencia binaria (RIB).

Este modelo evitando el empleo de fórmulas matemáticas, se basa en las siguientes consideraciones:

1. Para caracterizar los documentos de la colección se emplean ciertos términos de indización.
2. Dada una necesidad informativa del usuario, existe un subconjunto de documentos de la colección que contiene exclusivamente los documentos relevantes con relación a esta.
3. Se parte exclusivamente de la presencia o ausencia de los términos en los documentos de la colección. Se trata, pues también, de un modelo binario, como el modelo booleano.
4. El usuario no sabe cuáles son los términos de indización que configurarían la consulta ideal. Tampoco sabe en qué medida los términos empleados en la consulta permiten discernir de los documentos relevantes y rechazar simultáneamente los documentos irrelevantes.
5. Actúa sobre los términos que configuran la consulta del usuario, ponderándolos, esto es, imponiéndoles un peso a cada uno de ellos, mayor cuanto mejor permita discernir los documentos relevantes de los irrelevantes, y menor en caso contrario. De esta manera se persigue que el modelo efectúe la recuperación de información incidiendo sobre todo en los mejores términos de entre todos los empleados por el usuario en la consulta, minimizando la importancia de aquellos otros términos que, aun figurando en la consulta, son malos términos del conjunto de respuesta ideal.
6. Como tampoco se puede saber a priori cuáles, de entre los términos que configuran la consulta, son buenos términos y cuáles no lo son, a este modelo no le queda otro remedio que considerar, para cada uno de los términos empleados en la consulta, la *"probabilidad de ser un buen término"* (probabilidad de que el término empleado en la consulta esté presente en un documento del conjunto de documentos relevantes en relación a la consulta) y simultáneamente, para ese mismo término, la *"probabilidad de ser un mal término"* (probabilidad de que ese mismo término esté presente en un documento del conjunto de documentos irrelevantes en relación a la consulta).

7. Como las probabilidades nombradas anteriormente son desconocidas inicialmente a la consulta, este modelo se ve en la necesidad de efectuar una hipótesis inicial sobre sus valores. La obligatoriedad de hacer esa hipótesis inicial sobre las *"probabilidades de ser un buen término o de ser un mal término"* para cada término de la consulta se considera el principal inconveniente de este modelo.

El modelo probabilístico es capaz de calcular el grado de similitud existente entre cada documento de la colección y la consulta ponderada, consiguiendo ordenar los documentos de la colección en orden descendente de probabilidad de relevancia en relación con la consulta. De esta manera el modelo supera el gran inconveniente del modelo booleano, la equiparación exacta. El modelo probabilístico, aun siendo un modelo binario, efectúa **equiparación parcial**, lo que le permite ordenar los documentos de la respuesta conforme a su probabilidad de relevancia. Ya que no puede ponderar los términos de la colección porqué es un modelo binario, la equiparación parcial es posible gracias a la ponderación de los términos empleados en la consulta.

Una de las grandes aportaciones del modelo probabilístico a la recuperación de información consiste en el fenómeno denominado retroalimentación por relevancia. Que consiste en la utilización de información generada en procesos de recuperación anteriores o durante el propio proceso de búsqueda para mejorar los resultados solicitando al usuario, tras una respuesta inicial a la consulta, que analice los documentos recuperados y valore cuáles son relevantes. Con esta información se imponen nuevos pesos a las *"probabilidades de ser buen término o de ser un mal término"* para cada término de la consulta, obteniéndose así una nueva respuesta de documentos ordenados por su probabilidad de relevancia, gracias a la información suministrada directamente por el usuario.

Actualmente hay muchos modelos de recuperación de información que emplean alguna variante de la retroalimentación por relevancia para mejorar y refinar los resultados de las consultas. Quizá la más conocida se base en la sugerencia al usuario de los resultados precedidos a la siguiente advertencia: *"Otros usuarios que adquirieron o preguntaron por ese documento también adquirieron o preguntaron por estos otros"*. Es una manera de emplear la información procedente, en este caso, de procesos de recuperación anteriores.

Una aplicación de esto muy utilizada se centra en predecir la siguiente palabra dada una cantidad de palabras anteriores. Que es muy útil, por ejemplo, en el software de reconocimiento de voz, donde uno debe decidir correctamente qué es lo que dice el hablante, incluso cuando la calidad de la señal es deficiente o hay mucho ruido de fondo.

2.3 Modelo Vectorial

Como se ha observado en las dos secciones anteriores, el modelo probabilístico supera al modelo booleano ya que el probabilístico efectúa equiparación parcial mientras que el modelo booleano efectúa equiparación exacta. Sin embargo, ambos siguen presentado una característica negativa: ni el modelo booleano ni el modelo probabilístico tienen en cuenta la frecuencia con la que aparecen los términos de indización dentro de los documentos, porqué ambos son modelos binarios.

Parece lógico pensar que, si en un documento aparece un término una vez, y en otro documento aparece ese mismo término veinte veces, consideremos que en el primer documento la importancia del término es menor que ese mismo término en el segundo documento. En consecuencia, a esa necesidad, surge un tercer modelo de recuperación de información, el modelo vectorial.

El modelo vectorial fue presentado por Salton[3] en 1975 y posteriormente asentado en 1983 por McGill[4]. Fue usado por primera vez por el sistema SMART de recuperación de información. Utiliza pesos no binarios para los términos de los documentos para así poder calcular el grado de similitud entre documentos y consultas.

Esto permite que el conjunto de documentos obtenidos como resultado de una consulta pueda ser ordenado por un ranking de relevancia.

El modelo se basa en estos tres principios:

- La equiparación parcial, que es la capacidad de ordenar los resultados de una búsqueda, basándose en el grado de similitud entre cada documento de la colección y la consulta.
- La ponderación de los términos en los documentos, no limitándose a señalar la presencia o ausencia de estos, sino ponderando cada término en cada documento con un número real que refleje su importancia en el documento.
- La ponderación de los términos es la manera que el usuario pueda asignar pesos a los términos de la consulta que reflejen la importancia de estos con relación a su necesidad informativa.

La ponderación de números reales, que son los pesos que representan al documento, se le denomina vector del documento, permitiendo su representación en el espacio vectorial y, en consecuencia, su tratamiento matemático. Es decir, en el modelo vectorial tanto un documento como la consulta se representan mediante un vector de pesos para determinar la representatividad de los documentos de la colección. La formulación del vector se representa de la siguiente forma, ver Figura 1.

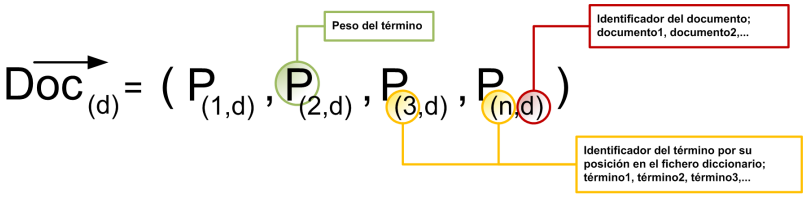

Figura 1: Representación del vector de un documento.

Donde n es el número total de términos considerados en la descripción de la colección y cada peso P es el producto de los valores TF-IDF, que se describen en la sección 3. Formalmente,

$$\mathrm{P}_{i,d} = \mathrm{TF}_{i,d} * \mathrm{IDF}_i$$

Gracias a esta representación, los documentos y las consultas se tratan matemáticamente como vectores en un espacio n dimensional y da el nombre al modelo. El modelo de espacio vectorial es posiblemente el modelo más exitoso e influyente para codificar palabras y documentos como vectores. Además, las representaciones vectoriales son requeridas en una amplia gama de algoritmos y métodos de aprendizaje automático que se utilizan para ayudar a abordar las tareas de PLN.

Veamos un ejemplo. Si consideramos únicamente dos dimensiones (dos únicos términos), concretamente los documentos D1 = (3, 5) y D2 = (4,1) y la consulta Q = (2,1) de un espacio bidimensional, se pueden dibujar tanto los documentos como la consulta en el plano de este documento, como *"flechas"* o vectores que parten del origen de coordenadas, cuyo primer número corresponde al valor del término 1 representado en el eje de abscisas y cuyo segundo número corresponde al valor del término 2 representado en el eje de ordenadas.

En este ejemplo obtendríamos el siguiente gráfico, donde D1 es el vector más próximo al eje de ordenadas, D2 es el vector más próximo al eje de abscisas, y donde la consulta Q es el vector entre D1 y D2:

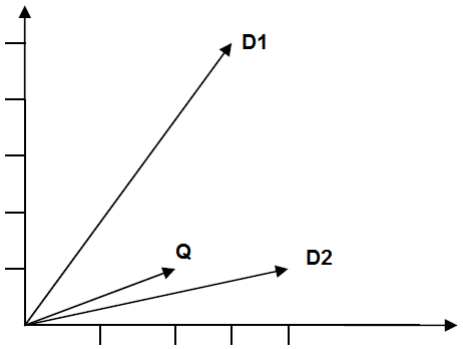

Figura 2: Ejemplo de representación del modelo vectorial.

Como podemos observar en el gráfico anterior (ver Figura 2), resulta relativamente fácil juzgar cuál de los dos documentos se asemeja más a la consulta. Considerando que el vector de la consulta Q está más próximo a D2, podemos deducir gráficamente que el orden de relevancia de los documentos D1 y D2 con relación a la consulta Q sería en este ejemplo D2 y posteriormente D1. Sin la ayuda de un gráfico bastaría fijar un criterio de similaridad para poder ordenar fácilmente por orden de relevancia los documentos de una colección con relación a una consulta (ver sección 4).

Las ventajas del modelo vectorial:

- Es un modelo simple basado en álgebra lineal.
- El peso de los términos no es binario.
- Permite aciertos parciales, ya que un documento puede ser considerado relevante, aunque no incluya todos los términos de la consulta.
- La ordenación de los resultados se realiza en base a varios factores: la frecuencia de los términos, la importancia de los términos y sin dar relevancia a los documentos más largos.
- Permite una implementación eficiente para grandes colecciones de documentos.

Las desventajas del modelo vectorial:

- Se pierde parte de la información sintáctica y semántica del documento, ya que se basa en la independencia de los términos dentro de un documento.
- Los documentos largos quedan poco representados ya que contienen pocos valores en común.
- El orden en el cual los términos aparecen en el documento se pierde en la representación espacio vectorial.
- Las palabras de búsqueda deben coincidir con las palabras del documento, partes de una palabra pueden dar falsos positivos.
- Sensibilidad semántica, documentos con contextos similares, pero con diferente vocabulario no serán asociados, resultando ser falsos negativos.

Las desventajas pueden ser solucionadas aplicando técnicas matemáticas, como la descomposición de un valor singular y bases de datos léxicas.

3 Medidas de cálculo de relevancia

Como se ha comentado anteriormente en la sección 2.3, las medidas de cálculo de relevancia típicamente usadas para la generación de los vectores son el factor término-frecuencia (TF) y el factor inverso de la frecuencia del documento (IDF). Que se usan para ponderar los términos en los documentos de la colección. Su multiplicación refleja la importancia de los términos.

El primer factor, TF (abreviatura de Term Frequency), pretende reflejar la importancia de los términos en los documentos, concediendo mayor importancia a los términos cuantas más veces aparezcan en los documentos. La versión más sencilla de este factor se representa numéricamente mediante la frecuencia de aparición de cada término en cada documento de la colección. A mayor frecuencia de un término en un documento, mayor importancia.

$$\mathrm{TF}_{d,i} = \frac{f_i}{max_j f_{d,j}}, \text{ donde } f_{d,j} \text{ es la frecuencia de } t_j \text{ en } d$$

El segundo factor, IDF (abreviatura de Inverse Document Frequency), pretende reflejar la importancia de los términos en la colección, dando más relevancia a la precisión y el poder discriminatorio de los mismos. Así, dará mayor importancia a un término cuanto menor sea el número de documentos de la colección en los que aparezca dicho término. Por el contrario, si un término aparece en todos los documentos de la colección, su precisión y poder discriminatorio (capacidad para discernir los documentos relevantes de los irrelevantes ante una consulta) será nulo (tal que el término aparecerá necesariamente tanto en todos los documentos relevantes como en todos los documentos irrelevantes), de manera que se le otorgará una importancia mínima en esa colección en concreto (puede que en otra colección ese mismo término posea una gran importancia, porque aparece en muy pocos documentos).

Suele representarse numéricamente de manera proporcional al logaritmo neperiano del inverso del número de documentos de la colección en los que aparece dicho término. Los términos más infrecuentes en la colección son más importantes, pues discriminan antes. Definimos la *"rareza"* de un término como su frecuencia inversa del documento, o IDF:

$$\mathrm{IDF}_i = log_2 \frac{D}{d_{f,i}},$$
donde D es igual al número de documentos y
$d_{f,i}$ es igual al número de documentos que contienen el termino t_i

4 Cálculo de similitud entre vectores de un MEV

El modelo vectorial propone evaluar el grado de similitud entre los documentos de una colección y las consultas mediante criterios que muestran la mayor o menor cercanía entre los vectores correspondientes a los documentos y el vector correspondiente a la consulta.

Una de las maneras más habituales de cuantificar el nivel de cercanía entre vectores es mediante el coseno del ángulo que forman, pues presenta la propiedad de ser un número mayor cuanto más cercanos estén entre sí ambos vectores, mientras que es un número menor cuanto más alejados estén entre sí.

A continuación, se muestran los criterios para evaluar el grado de similitud entre vectores de un MEV más conocidos.

4.1 Mediante el producto escalar

Como veremos, existen muchas modalidades de comparación o equiparación mediante el grado de similitud. Una de las más sencillas por su simplicidad y sistematización inmediata es mediante el producto escalar de los pesos.

En el cálculo de similitud mediante el producto escalar, la similitud de un documento y una consulta, seran igual a la suma de los productos de sus pesos, sin olvidar que cada peso representa a un término. Este método puede aplicarse tanto a pesos binarios como a pesos TF-IDF.

4.1.1 Modalidad de pesos binarios

En el caso de la modalidad de pesos binarios (ver Figura 3), la similitud de un documento con respecto a la consulta es equivalente a la presencia de los términos de la consulta en el documento. Esto quiere decir que la ausencia de un término de la consulta o del documento implica un producto igual a 0 y por lo tanto no tienen incidencia en el cálculo. Por el contrario, la presencia de un término tanto en la consulta como en el documento siempre tendrá valor 1. Para ello sólo bastará contabilizar el número de términos coincidentes de la consulta en el documento y ése será su valor de similitud.

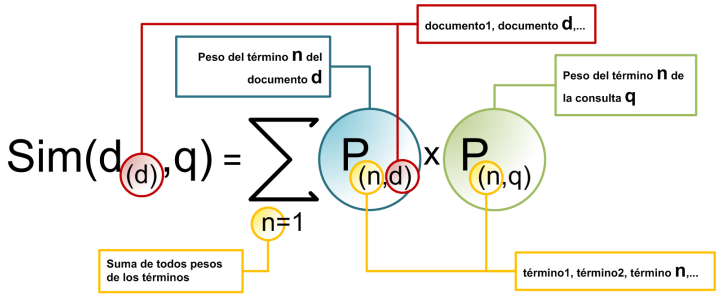

Figura 3: Similitud de un documento d y la consulta q mediante producto escalar.

4.1.2 Modalidad de pesos TF-IDF

En el caso de la modalidad anterior de pesos binarios, las limitaciones en la definición de la representatividad de los términos de cada documento son patentes. Por tanto, es un resultado bastante limitado y parcial. Por ello el método de la similitud mediante el producto escalar se aplica habitualmente con pesos TF-IDF, mucho más precisos.

El cálculo de la similitud se aplica a cada uno de los documentos de la colección. Por ejemplo, para el primer documento la similitud con respecto a la consulta de un usuario, será diferente respecto al segundo documento. Al igual que ocurría con los pesos binarios, sólo tienen incidencia aquellos términos presentes tanto en la consulta como en el documento, ya que sus pesos se multiplican y se suman sucesivamente al resto, siendo estos mucho más precisos que un simple número entero.

Ejemplo de la matriz A (ver Tabla 1) con pesos TF-IDF:

	D1	D2	D3
T1	0.176	0.000	0.000
T2	0.000	0.000	0.176
T3	0.176	0.954	0.176

Tabla 3: Matriz A con pesos TF-IDF.

4.2 Mediante la fórmula del coseno

Mediante el producto escalar, el proceso de equiparación es posible cuando en el vector de la consulta y en el del documento existen términos coincidentes. Pero este enfoque no supone la representación del vector de la consulta y del documento. De hecho una de las claves del modelo de espacio vectorial es precisamente la posibilidad de determinar el ángulo que forman los vectores del documento y de la consulta que se está comparando (ver Figura 4).

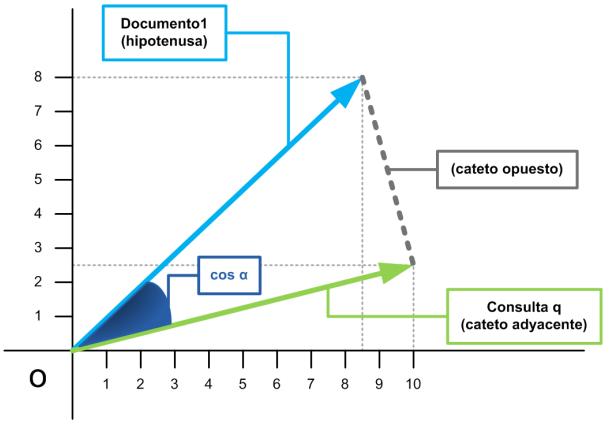

Figura 4: El ángulo del coseno.

Es posible medir cuál es la desviación de un documento con respecto a una consulta, por el número de grados del ángulo que forman porque crean una estructura triangular a la que se aplica el cálculo del ángulo que forma la hipotenusa (el vector del Documento1) y el adyacente (el vector q de la consulta dada por el usuario) que resulta ser el coseno del triángulo.

En el caso de la Figura 4, se comprueba visualmente cierta distancia del vector de la consulta con respecto al Documento1, cuando ambos vectores se muestran tan próximos como para superponerse, implica que el ángulo que forman sea menor y que su nivel de coincidencia sea superior. De hecho, el coseno de alfa de un triángulo cualquiera siempre es igual al cateto adyacente entre la hipotenusa y un coseno de 0^{o} implica una similitud máxima.

Por tanto, la fórmula aplicada para calcular el coeficiente de similitud del coseno (ver Figura5) entre un documento y una consulta es aquella que permite poner en relación los pesos que forman los vectores del documento y la consulta. El numerador no deja de ser un producto escalar entre los pesos del documento y la consulta, y el denominador la raíz cuadrada del producto del sumatorio de los pesos del documento y la consulta al cuadrado. La formulación del denominador con raíz cuadrada y cálculo de cuadrados se diseñó para conseguir un resultado final de la división, inferior a 1, de tal manera que el coeficiente fuera fácil de calcular.

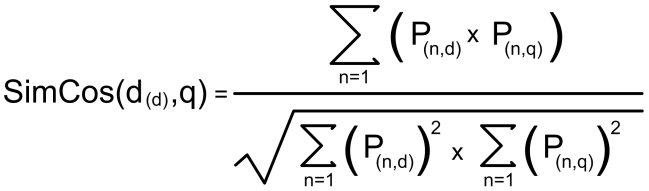

Figura 5: Fórmula para el cálculo de la similitud del coseno.

4.3 Mediante el coeficiente de Dice

El cálculo del coeficiente de similitud según Lee Raymond Dice[5] es una adaptación del cálculo del coeficiente del coseno. La diferencia en la formulación reside en que la cardinalidad del numerador es dos veces la información compartida y el denominador la suma de los pesos al cuadrado del documento y su consulta.

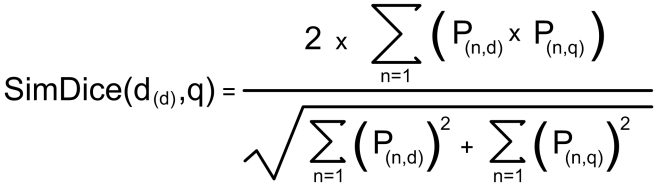

Figura 6: Fórmula para el cálculo del coeficiente de similitud de Dice.

4.4 Mediante el coeficiente de Jaccard

El cálculo del coeficiente de similitud de Jaccard[6] al igual que el de Dice[5], son una adaptación del coeficiente de similitud del coseno.

La aplicación del coeficiente de similitud de Jaccard, centrada en usos estadísticos, mide la similitud entre conjuntos. Se puede definir como el tamaño de la intersección (numerador) dividido por el tamaño de la unión de la muestra, en este caso la suma de los pesos al cuadrado del documento y la consulta menos la intersección.

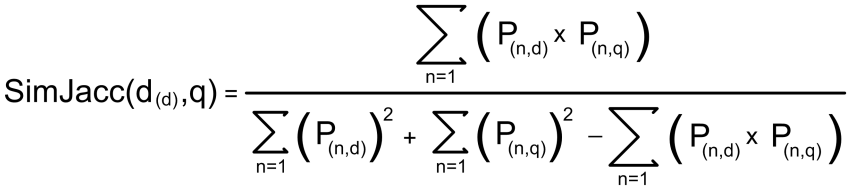

Figura 7: Fórmula para el cálculo del coeficiente de similitud de Jaccard.

Una vez calculada la similitud entre cada documento de la colección y la consulta, el modelo vectorial es capaz de ordenar todos los documentos de la colección en orden decreciente de su grado de similitud con la consulta, incorporando a los resultados aquellos documentos que satisfacen sólo parcialmente los términos de la consulta. Se efectúa, en consecuencia, equiparación parcial.

5 Espacios vectoriales basados en *word embeddings*

Los espacios vectoriales basados en representaciones continuas o densas de palabras se denominan comúnmente *word embeddings*[9]. Estos codifican información sintáctica y semántica sorprendentemente bien y han demostrado ser muy útiles como características adicionales en muchas de las tareas de PLN. Concretamente, la tarea de representar palabras y documentos es parte integrante de la mayoría, si no de todas, las tareas de PLN.

Como hemos comentado en las secciones anteriores, se ha demostrado que es muy útil la representación de las palabras y los documentos como vectores, ya que tienen una interpretación más atractiva e intuitiva, y pueden ser objeto de operaciones útiles. Además, se usan en muchos algoritmos y estrategias de Machine Learning.

Las características principales de los *word embeddings* es que son densos, distribuidos, vectores de palabras de longitud fija y se construyen utilizando estadísticas de coincidencia de palabras según la hipótesis de distribución.

Con el tiempo, los *word embeddings* se han convertido en un tema de investigación en sí mismos, al darnos cuenta de que pueden usarse como características independientes en muchas tareas de PLN y el hecho de que codifican de manera precisas relaciones de palabras sintácticas y semánticas.

La investigación moderna sobre los *word embeddings* se basa particularmente en modelos basados en predicciones, fruto de los intentos de hacer que las tareas de PLN sean más eficientes y precisas.

Inicialmente la creación de los *word embeddings* se basaba en redes neuronales y capas de incrustación, pero más recientemente, han surgido otras formas de crear *word embeddings*, que no se basan en redes neuronales y capas de incrustación, sino que aprovechan las matrices de contexto de palabras para llegar a representaciones vectoriales de palabras.

Como se mencionó en las características de los *word embeddings*, los podemos considerar como representaciones vectoriales de longitud fija de palabras. Hay varias formas de obtener tales representaciones y los *word embeddings* son comúnmente clasificados dependiendo de las estrategias utilizadas en los dos métodos siguientes:

- En métodos que aprovechan los datos locales (por ejemplo, el contexto de una palabra), que se denominan modelos basados en predicciones. Su objetivo es modelar la probabilidad de la siguiente palabra dada una secuencia de palabras y que aprovechan las estadísticas de coincidencia global.
- Por otro lado, en métodos que usan información global, generalmente estadísticas de todo el corpus, como el recuento de las palabras y las frecuencias, que se denominan modelos basados en conteo.

Se ha encontrado que los *word embeddings* son muy útiles para muchas tareas de PNL, incluidas, entre otras, chunking, preguntas y respuestas y análisis de opinión. Muchas de sus aplicaciones han sido su incorporación en kits de herramientas ampliamente utilizadas como Word2Vec, Doc2vec, FastText y GloVe (ver sección 6).

6 Algunas de las aplicaciones de los *word embeddings*

Al inicio del documento se ha comentado que una de las investigaciones fundamentales en el desarrollo de los MEV son las aplicaciones en diversas tareas de clasificación textual mediante el uso de representaciones continúas de palabras o vectores densos de palabras (*word embeddings*).

A continuación, se darán breves introducciones teóricas de las aplicaciones acabadas de nombrar al final de la sección 5.

6.1 Word2vec

Word2vec[7] es un grupo de modelos relacionados que se utilizan para producir incrustaciones de palabras. Estos modelos son redes neuronales superficiales de dos capas que están entrenadas para reconstruir contextos lingüísticos de palabras. *Word2vec* toma como entrada un gran corpus de texto y produce un espacio vectorial, típicamente de varios cientos de dimensiones, a cada palabra única en el corpus se le asigna un vector correspondiente en el espacio. Los vectores de palabras se colocan en el espacio vectorial de tal manera que las palabras que comparten contextos comunes en el corpus se ubican cerca unas de otras en el espacio.

Word2vec fue creado, publicado y patentado en 2013 por un equipo de investigadores en Google dirigido por Tomas Mikolov. El algoritmo ha sido analizado y explicado posteriormente por otros investigadores. Los vectores incorporados creados con el algoritmo *Word2vec* tienen muchas ventajas en comparación con otros algoritmos, como el análisis semántico latente.

6.2 Doc2vec

Doc2vec[8] es un algoritmo no supervisado para generar vectores para oraciones, párrafos y documentos. El algoritmo es una adaptación de *Word2vec* que puede generar vectores para palabras. Los vectores generados por *Doc2vec* se pueden usar para tareas PLN, como encontrar similitudes entre oraciones, párrafos y documentos.

A diferencia de otros modelos, donde la secuencia de palabras se captura en los vectores de oración generados, los vectores de oración *Doc2vec* son independientes del orden de las palabras.

Para las tareas de similitud de oraciones, los vectores *Doc2vec* pueden funcionar razonablemente bien. Sin embargo, si el corpus de entrada contiene muchos errores ortográficos, este algoritmo puede no ser la opción ideal. Puede ser mejor generar vectores de palabras construidos a partir de n caracteres usando fastText.

Hay una implementación llamada Gensim para *Doc2vec*.

6.3 FastText

FastText es otro algoritmo que generar vectores de palabras construidos a partir de nn caracteres y luego suma esos vectores de palabras para componer un vector de oración. También es un algoritmo no supervisado para generar vectores donde el vector para una oración se genera prediciendo oraciones adyacentes, que se supone que están semánticamente relacionadas.

Cuenta con una biblioteca para el aprendizaje de los *word embeddings* y la clasificación de texto creada por el laboratorio de investigación de Inteligencia Artificial (FAIR) de Facebook. Facebook pone a disposición modelos preentrenados para 294 idiomas. *FastText* utiliza una red neuronal para incrustar palabras.

6.4 GloVe

GloVe, que recibe el nombre de Global Vectors, es un modelo para la representación de palabras distribuidas. El modelo es un algoritmo de aprendizaje no supervisado para obtener representaciones vectoriales de palabras. Esto se logra mapeando las palabras en un espacio significativo donde la distancia entre las palabras está relacionada con la similitud semántica. La capacitación se lleva a cabo en estadísticas globales de coincidencia global palabra-palabra de un corpus, y las representaciones resultantes muestran estructuraras lineales interesantes del espacio vectorial de palabras.

Se desarrolla como un proyecto de código abierto en Stanford. Como modelo de regresión lineal para el aprendizaje no supervisado de representaciones de palabras, combina las características de dos familias de modelos, la factorización matricial global y los métodos de ventana de contexto local.

GloVe se puede usar para encontrar relaciones entre palabras como sinónimos, relaciones de empresa-producto, códigos postales y ciudades, etc. También se usa en el modelo SpaCy para construir incrustaciones de palabras semánticas y vectores de características al calcular las palabras de la lista superior que coinciden con la distancia medidas como la similitud de coseno y el enfoque de distancia euclidiana. También se utiliza como marco de representación de palabras para los modelos en línea y fuera de línea diseñados para detectar la angustia psicológica en las entrevistas con pacientes.

7 Reflexión personal sobre la utilidad y futuro de los MEV

Actualmente la recuperación de información ha cobrado un gran auge debido al crecimiento espectacular de Internet, tratando de facilitar la tarea de comprensión de los escasos documentos relevantes que puedan existir en la red frente a los millones de documentos irrelevantes con relación a cada consulta formulada en la red. Dado que esta inmensa *"colección"* carece por completo de organización, la automatización de los procesos de análisis y recuperación de los billones de documentos que configuran la red se ha convertido en una tarea de gran importancia.

Los programas que rastrean la web en busca de páginas y los programas que efectúan el proceso de análisis y tratamiento de tales páginas con el objeto de poder recuperar información ante las consultas de los usuarios, además de muchos otros programas con un objetivo semejante en cualquier ámbito (desde las bibliotecas hasta el comercio electrónico), se siguen basando en los tres modelos clásicos de recuperación de información creados entre los años sesenta y ochenta: los modelos booleano, probabilístico y vectorial.

Como hemos observado, el fenómeno más destacado actualmente en estos modelos de recuperación de información consiste en el empleo simultáneo de características y algoritmos propios de cada uno de estos modelos. Así, lo más frecuente es que los buscadores de Internet se basen en el modelo booleano, pero efectúen la ordenación de los documentos de las respuestas empleando criterios de similitud originarios del modelo vectorial. De igual modo, cada vez en mayor medida los modelos emplean una u otra variante de la retroalimentación por relevancia para aumentar la precisión de la respuesta, técnica empleada en sus inicios por el modelo probabilístico.

En consecuencia, puede afirmarse que con la popularización de Internet han cobrado mucha importancia los modelos MEV de recuperación de información, tratando de unir en un mismo programa de recuperación las ventajas primordiales de cada uno de ellos.

La investigación en esta área, muy activa en la actualidad, sigue tratando de mejorar la precisión y exhaustividad de los modelos de recuperación de información, pero tratando ahora de incorporar el usuario real y su punto de vista subjetivo en la evaluación de los modelos. Sin duda en un futuro, tales avances se incorporarán en beneficio de un acceso rápido y eficaz a la información por parte de cualquier usuario.

Bibliografía

[1] Robertson, S. E. *The probability ranking principle in IR.* Journal of Documentation, 1977, 33(4):294-304.

[2] Sparck Jones, K. *Search term relevance weighting given little relevance information.* Journal of Documentation, 1979, 35(1):30-48.

[3] Salton, G.; Wong, A.; Yang, C.S. *A vector space model for automatic indexing.* Communications of the ACM, vol. 18, nr. 11, pp. 613–620.

[4] Salton, G.; McGill, M.J *Introduction to Modern Information Retrieval.* Mc Graw Hill, 1983.

[5] Dice, Lee R. *Measures of the Amount of Ecologic Association Between Species.* Ecology 1945, 26(3):297-302.

[6] Real, R.; Vargas, J. M. *The probabilistic basis of Jaccard's index of similarity.* Systematic biology 1996, 45(3), 380-385.

[7] Tomas Mikolov; Ilya Sutskever; Kai Chen; Greg Corrado; Jeffrey Dean. *Distributed Representations of Words and Phrases and their Compositionality.* CoRR 2013, `http://arxiv.org/abs/1310.4546`.

[8] Quoc V. Le; Tomas Mikolov. *Distributed Representations of Sentences and Documents.* CoRR 2014, `http://arxiv.org/abs/1405.4053`.

[9] Felipe Almeida; Geraldo Xexéo. *Word Embeddings: A Survey.* CoRR 2019, `http://arxiv.org/abs/1901.09069`.