**Introducción: → Vicent**

La medición fiable de la similitud de textos es una aplicación de gran ayuda para distintas tareas relacionadas con el procesamiento de textos en lenguaje natural. Solo por citar algunos ejemplos, los sistemas de clasificación de documentos, de resúmenes de textos, o de traducción automática, pueden beneficiarse de un módulo de similitud que establezca el grado de parecido entre dos unidades textuales. En general, podemos distinguir entre dos tipos de similitud: a nivel de palabras y a nivel de textos.

Las distintas aproximaciones de similitud a nivel de palabras se agrupan en dos categorías: similitud léxica de palabras y similitud semántica de palabras. Dos palabras son similares a nivel léxico si están compuestas por secuencias parecidas de caracteres. La similitud semántica de palabras, por su parte, nos permite medir si dos palabras tienen significados parecidos o se usan en contextos parecidos. Hay dos grandes grupos de técnicas para calcular la similitud semántica de palabras: técnicas basadas en conocimiento y técnicas basadas en corpus.

Las técnicas de similitud de palabras basadas en corpus determinan el parecido semántico de dos palabras en función de los usos de esas palabras en una gran colección de textos.

Muchas de las aproximaciones de similitud de textos se basan en la idea de alineamiento, que básicamente consiste en un emparejamiento entre palabras de los dos textos a comparar.

**Definición de WORD EMBEDDING → Vicent**

Word embedding es el nombre de un conjunto de lenguajes de modelado y técnicas de aprendizaje en procesamiento del lenguaje natural (PLN) en dónde las palabras o frases del lenguaje natural son representadas como vectores de números reales.​

word embeddings es que son no supervisadas

Las técnicas de word embedding parten de representaciones BOW (bag of words) de los distintos contextos de las palabras para obtener representaciones vectoriales de las palabras de dimensiones mucho más reducidas que capturan el significado y las relaciones entre palabras. Hay diversas técnicas para calcular estas representaciones, una de las más empleadas se basa en redes neuronales de una sola capa oculta que predicen la palabra dado el contexto o viceversa, adaptando así una de las piezas básicas de los modelos de aprendizaje profundo, los autocodificadores.

**WORD2VEC → Luis**

creada y patentada en 2013 por un equipo de Google dirigido por tomas mikolov director de proyecto

Esta tecnica hay sido mejorada en estos ultimos años y La mayoría de las nuevas técnicas utilizan arquitecturas de redes neurales

La mayoría de las nuevas técnicas utilizan arquitecturas de redes neurales en vez de los modelos más tradicionales como los n-gramas y el aprendizaje no supervisado.​

Estas herramiendtas hoy en dia consisten en odelos son redes neuronales superficiales de dos capas que se entrenan para reconstruir los contextos lingüísticos de las palabras.11º1

Word2vec toma como entrada un gran [corpus de texto](https://en.wikipedia.org/wiki/Text_corpus) y produce un [espacio vectorial](https://en.wikipedia.org/wiki/Vector_space) ,que normalmente tiene cientos de [dimensiones](https://en.wikipedia.org/wiki/Dimensions) , y a cada palabra única le asigna un vector correspondiente en el espacio. Estos vectores se colocan en el espacio vectorial de manera que las palabras que comparten contextos comunes en el corpus se ubican cerca unas de otras en el espacio

**Lattent SsemantingAnalisis: → Luis**

Analiza las relaciones entre un conjunto de documentos y los términos que contienen, estableciendo que dos palabras son similares si ocurren en fragmentos similares de textos. LSA parte de una matriz de palabras frente a documentos y aplica una técnica matemática denominada singular value decomposition que permite reducir el número de filas (documentos) preservando la similitud entre columnas (palabras).º

**CBOW Model (Common Bag of Words): → Luis**

Este modelo coge el entrada contexto de cada palabra como e intenta predecir la palabra correspondiente al contexto.

Como entrada se pueden pasar múltiples palabras de contexto La entrada o la palabra de contexto es un vector codificado de longitud , y la capa oculta calcula la media de todas ellas.

*V.* La capa oculta (hidden layer) contiene *N* neuronas y la salida es otra vez un vector de longitud *V* con los elementos máximos suavizados. La capa oculta copia la suma ponderada de las entradas a la capa siguiente.

El modelo de *Skip-Gram* es lo contrario al *CBOW,* tiene como entrada una palabra objetivo y como salida obtiene *C* distribuciones de probabilidades.

El CBOW es más rápido y tiene mejores representaciones para las palabras más frecuentes.

(Diego: INICIO)

**Agregación:**

* El método de agregación nos permite construir representaciones vectoriales de los textos a partir de los vectores individuales de palabras.
* Una vez obtenido el vector agregado para cada texto se aplican medidas de similitud ‘tradicionales’, como son la distancia euclídea y la similitud del coseno
* Tras probar distintas funciones de agregación, se eligió como representación la media aritmética de los vectores, obtenida sumando todos los vectores de palabras y dividiendo entre el número de palabras que componen el texto. ​

**Alineamiento:**

El método de alineamiento se apoya en la similitud de vectores de palabras para determinar la vinculación entre las mismas.

El alineamiento entre dos textos proporciona un marco sobre el que se pueden evaluar distintas heurísticas para determinar el grado de similitud entre los textos. Por ejemplo, usando las conexiones propuestas en el alineamiento para calcular métricas de similitud semántica entre las palabras emparejadas y agregando estas métricas individuales para obtener una métrica de similitud global entre los dos textos.

El proceso comienza con la construcción de un vector en el que existe una posición asociada a cada palabra del conjunto formado por la unión de los dos textos a analizar. Para cada palabra del primer texto se buscará una palabra del segundo texto con la que alinearla. Si la misma palabra existe en los dos textos se alineará consigo misma, y en caso contrario se calculará la distancia entre las representaciones vectoriales de la palabra de referencia del primer texto y todas las del segundo, con el fin de buscar la más cercana. El valor a introducir en la posición del vector asociada a la palabra de referencia será un 1 en el primer caso, y 1 menos la distancia en el segundo. Si el alineamiento no se puede llevar a cabo se introducirá un 0. Esto puede suceder si la palabra no está presente en el vocabulario del modelo de word embeddings utilizado.

(Diego: FIN)

**Traducción automática: → David**

En este primer modelo de similitud de textos entre distintos idiomas procederemos a la traducción automática del texto al idioma opuesto, de forma que para cada frase tendremos su representación en los dos idiomas (español - inglés). Con estas representaciones vectoriales haremos uso de las técnicas antes mencionadas de agregación y alineamiento para calcular su similitud.

**Transformación de modelos: → David**

La segunda aproximación a la hora de calcular la similitud entre textos de distintos idiomas se basa en la idea propuesta en (Mikolov, Le, y Sutskever, 2013), que dice lo siguiente: conceptos similares, expresados en distintos idiomas, deben tener distribuciones geométricas similares en un espacio vectorial. De esta forma, dados dos modelos entrenados uno en cada lenguaje, es posible aprender una matriz de transformación lineal entre ambos modelos.

**Bert: → David**

BERT

* "Google, búscame: Título de la película donde el protagonista pueda ver el futuro con unos lentes".​
* Esto antes no funcionaba.​
* Con la ayuda de la IA, Google creó BERT.​
* Gracias a este sistema, Google sabe discernir mejor el contexto de las palabras en las consultas de búsqueda.

Word2Vec vs. BERT

* Word2Vec genera *embeddings* que son **independientes del contexto**: es decir, solo hay una representación vectorial por palabra.​
* BERT genera *embeddings* que permiten tener **múltiples representaciones vectoriales para la misma palabra**, en función del contexto en el que se utiliza la palabra. Por tanto, los *embeddings* de BERT dependen del contexto.​
* BERT será capaz de distinguir y capturar los dos significados semánticos diferentes produciendo dos vectores diferentes para la misma palabra "bank".