TEMA 9. OTROS MODELOS EN RECUPERACIÓN DE INFORMACIÓN

Contenidos

En los temas anteriores se han estudiado algunos modelos clásicos de RI, el modelo booleano y el modelo vectorial. En este último tema conoceremos los modelos probabilísticos.

En todos los modelos estudiados, un término es un elemento de un conjunto, el vocabulario de términos, o bien se representa como un vector de dimensión la talla de la colección de documentos con unos pesos que dependen generalmente de la frecuencia del término en la colección de documentos.

En este tema vamos a introducir el concepto general de representación vectorial de los términos calculada en base al contexto de aparición de dichos términos. De esta forma se puede plantear buscar similitudes semánticas entre términos o entre consultas y documentos.

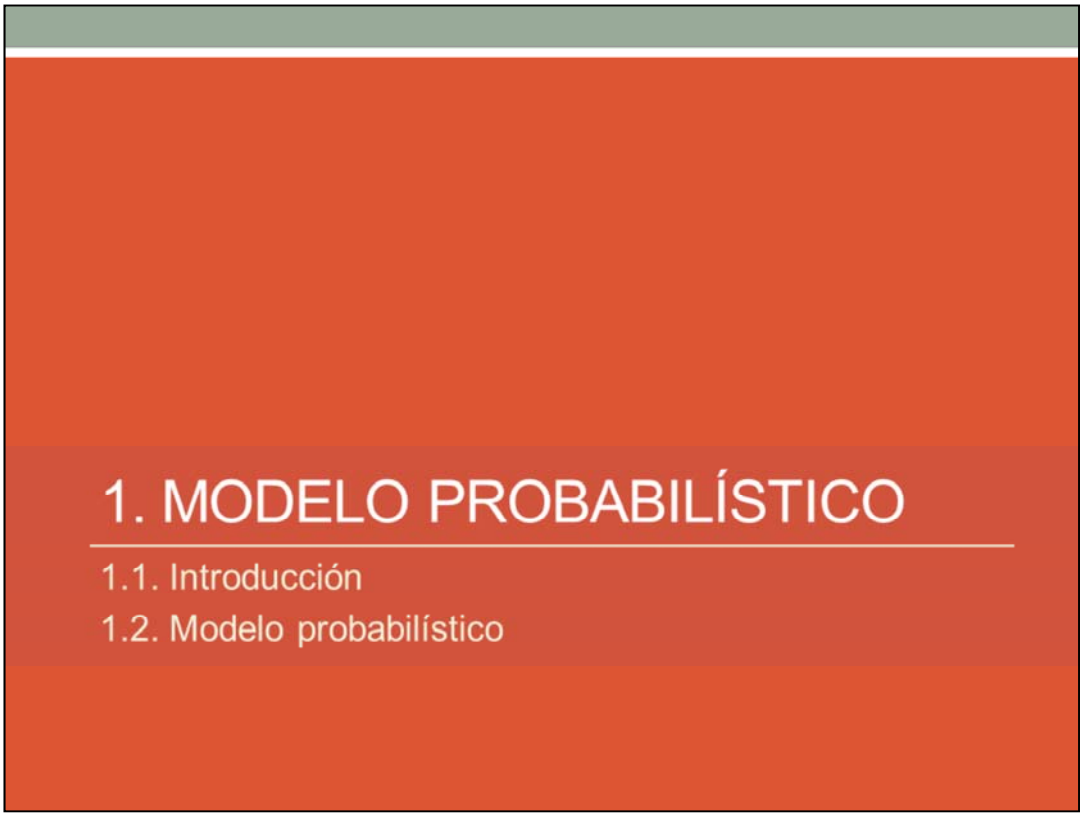
Finalmente, introduciremos el uso de redes neuronales como parte de un sistema de RI.

Bibliografía

A Introduction to Information Retrieval:
Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze.
Cambridge University Press, ***2009****. Capítulo 11*

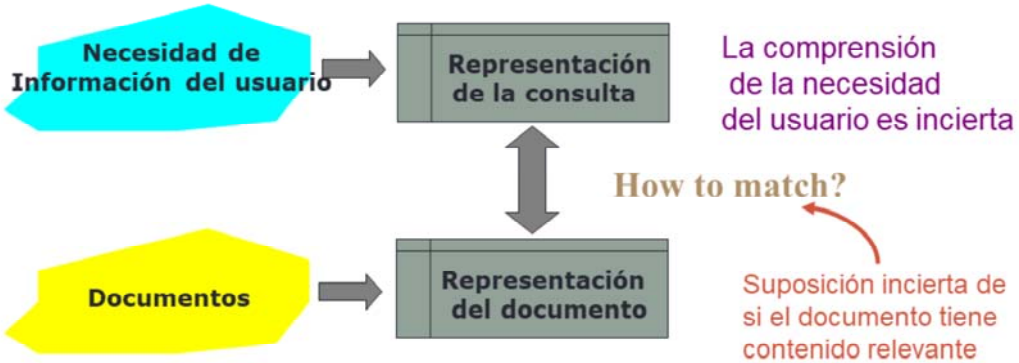
Speech and Language Processing
Daniel Jurafsky, James H. Martin.
Third Edition draft https://web.stanford.edu/~jurafsky/slp3/
2018. Capítulo 6

Neural Text Embeddings for Information Retrieval
Bhaskar Mitra, Nick Craswell
Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017



Vamos a plantear cómo puede la teoría de las probabilidades contribuir al diseño de un sistema de RI

1.1 Introducción:
¿por qué probabilidades en RI?



- En sistemas tradicionales de RI, el matching entre cada documento y la consulta se realiza en el espacio semántico impreciso de los términos.
- La teoría de la probabilidad proporciona una base de principios para el razonamiento incierto.
- *¿Podemos usar probabilidades para cuantificar nuestras incertidumbres?*

1.1 Introducción:
el problema del puntuación de documentos

- Dada una colección de documentos y una consulta del usuario, el sistema debe devolver una lista de documentos.
- **Los métodos de puntuación son el núcleo de un Sistema de RI.**
- **Idea:**
  - **Utilizar la probabilidad de relevancia de cada documento con respecto a la necesidad de información.**
    - P(R=1|document, query)

1.2 Modelo probabilístico

- Los resultados de recuperación de un modelo probabilístico dependen de la estimación de probabilidades.
- La primera asunción es que los términos están distribuidos de formas diferentes en los documentos relevantes y en los no relevantes.
- Un modelo probabilístico puntúa y ordena los documentos en orden decreciente de probabilidad de relevancia para la información requerida por el usuario, una vez las probabilidades han sido calculadas.
- En la fase de recuperación a cada documento se le asigna un valor que corresponde a la suma de probabilidades a partir de los términos comunes entre el documento y la consulta.

1.2 Modelo probabilístico:

Binary Independence Model: Se asume que cada término ocurre en cada documento de forma independiente.

De forma similar al modelo vectorial, se crea un vector que refleja la importancia de cada término.

Sea p_i la probabilidad de que un documento que contiene un término *i* sea relevante para una consulta.

Sea s_i la probabilidad de que un documento que contiene el término *i* sea no relevante para la consulta.

La puntuación se calcula como:

$$\sum_{i:d_i=1} \log \frac{p_i(1-s_i)}{s_i(1-p_i)}$$

p_i = número de documentos relevantes que contienen el término *i* / número total de documentos relevantes

s_i = número de documentos no relevantes que contienen el término *i* / número total de documentos no relevantes

1.2 Modelo probabilístico

Sean:

- n_i = número de documentos que contienen el término *i*
- r_i= número de documentos relevantes que contienen el término *i*
- N = número total de documentos
- R = número de documentos relevantes

Podemos expresar p_i y s_i como:

	Relevant	Non-relevant	Total
$d_i = 1$	r_i	$n_i - r_i$	n_i
$d_i = 0$	$R - r_i$	$N - n_i - R + r_i$	$N - r_i$
Total	R	$N - R$	N

1.2 Modelo probabilístico

La función de puntuación queda como:

$$\sum_{i:d_i=q_i=1} \log \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}$$

Se añade un factor 0,5 a cada componente para evitar los ceros en el denominador.

13

1.2 Modelo probabilístico: (Okapi) BM25

Popular y efectivo algoritmo de puntuación que incorpora la frecuencia de los términos en documento (2º factor) y consulta (3r factor).

$$\sum_{i \in Q} \log \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)} \cdot \frac{(k_1+1)f_i}{K+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

f_i y qf_i son las frecuencias del término en documento y consulta respect.
k_1, k_2 y K son parámetros que se fijan empíricamente
TREC: k_1 = 1.2, k_2 varía de 0 a 1000, b = 0.75

$$K = k_1((1-b) + b \cdot \frac{dl}{avdl})$$

para controlar el efecto de las diferentes longitudes de los documentos

dl es la longitud del documento
avdl la longitud media de los documentos de la colección

El primer factor se aproxima cuando el sistema no tiene datos para su estimación con una variante del componente idf del modelo vectorial.

Vamos a plantear cómo representar las palabras de un vocabulario como vectores en un espacio vectorial.

Para ello recuperaremos el concepto del modelo vectorial en recuperación de información.

2.1 Las palabras y su contexto

- Las palabras que aparecen en contextos similares tienden a tener significados similares.
- Este vínculo entre la similitud en la forma en que se distribuyen las palabras en los textos y la similitud en lo que significan se llama hipótesis distribucional.

2.1 Las palabras y su contexto

Hipótesis distribucional:

Zellig Harris (1954):

- Las palabras “oculista” y “oftalmólogo” … suelen ocurrir en los mismos contextos
- Si A y B aparecen con frecuencia en los mismos contextos decimos que son sinónimos

Firth (1957):

- “You shall know a word by the company it keeps!”

2.1 Las palabras y su contexto

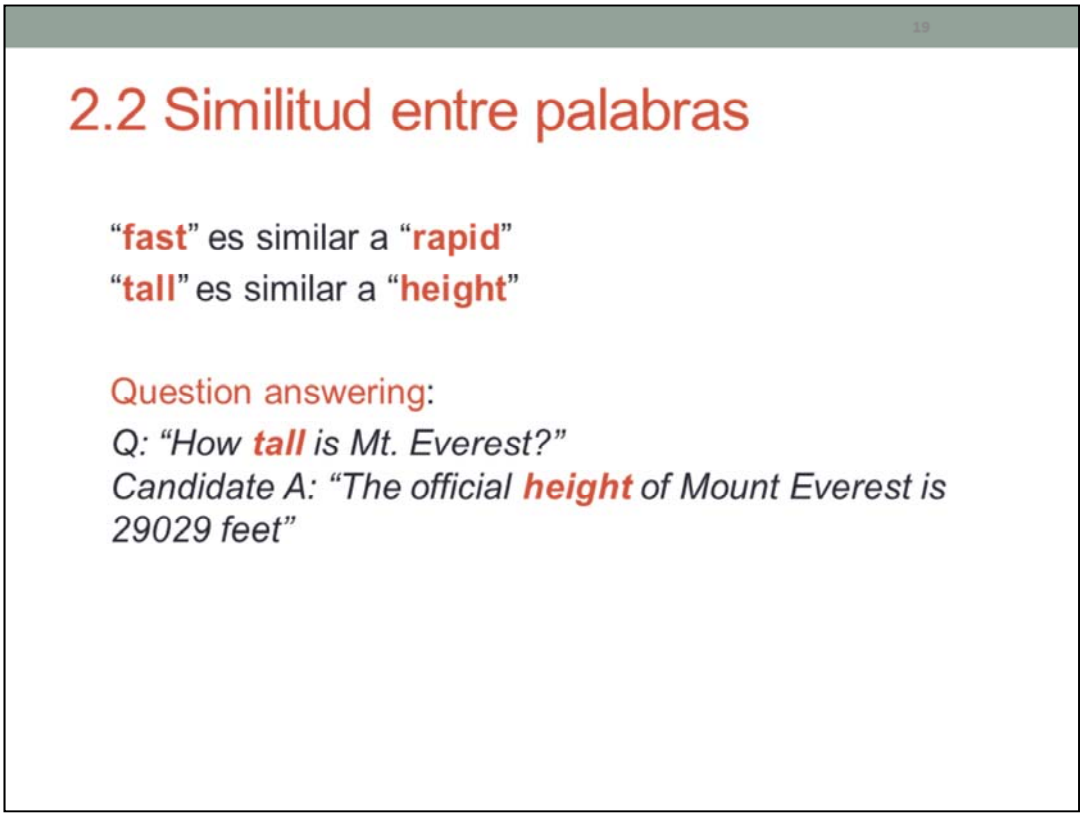
Ejemplo:

A bottle of ***tesgüino*** is on the table
Everybody likes ***tesgüino***
Tesgüino makes you drunk
We make ***tesgüino*** out of corn.

Por las palabras del contexto los humanos podemos conjeturar que el significado de ***tesgüino*** es una bebida alcohólica como la cerveza.

2.2 Similitud entre palabras

- Tradicionalmente en el procesamiento del lenguaje natural una palabra se ha representado como un elemento de un conjunto (un índice en un diccionario).
- Sin embargo, para múltiples aplicaciones, por ejemplo, el cálculo de la similitud semántica de textos, se hace necesario establecer distancias o similitudes entre palabras.



La similitud entre 'tal' y 'heigth' permitiría asignar una puntuación alta a 'Candidate A' como respuesta a la pregunta 'Q':

2.2 Similitud entre palabras

Idea: representar las palabras en un espacio vectorial de forma que cada palabra tendrá asociado un vector, y por tanto, un punto en ese espacio.

- Se modela el significado de una palabra embebiéndolo en un espacio vectorial.
- El 'significado' de una palabra es un vector de números

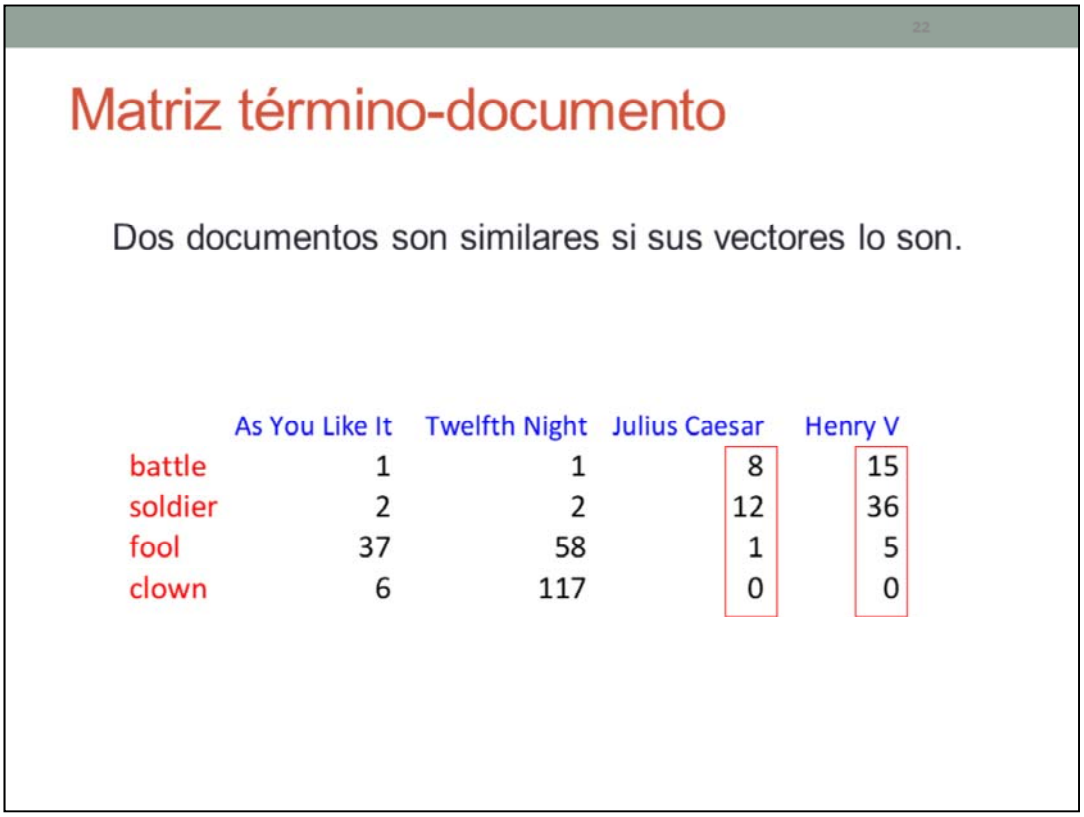
  Algunas representaciones vectoriales de las palabras suelen llamarse "**embeddings**".

Matriz término-documento

- En cada celda se guarda el contador de ocurrencias del término t en el documento d, ($f_{t,d}$).
- Cada documento se representa por un vector en $\mathbb{N}^{|V|}$, una columna de la matriz. Donde V = nº de términos diferentes de la colección

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Partimos del mismo modelo que se ha estudiado para el modelo vectorial de RI, recuperamos el concepto de matriz de ocurrencia término-documento. En este modelo representamos vectorialmente los documentos en base a las columnas de la matriz.



Establecemos similitudes entre documentos en base a la similitud de sus vectores.

Las palabras en la matriz término-documento

Cada palabra es un vector en $\mathbb{N}^{|N|}$, una fila de la matriz.

Donde N = número de documentos.

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

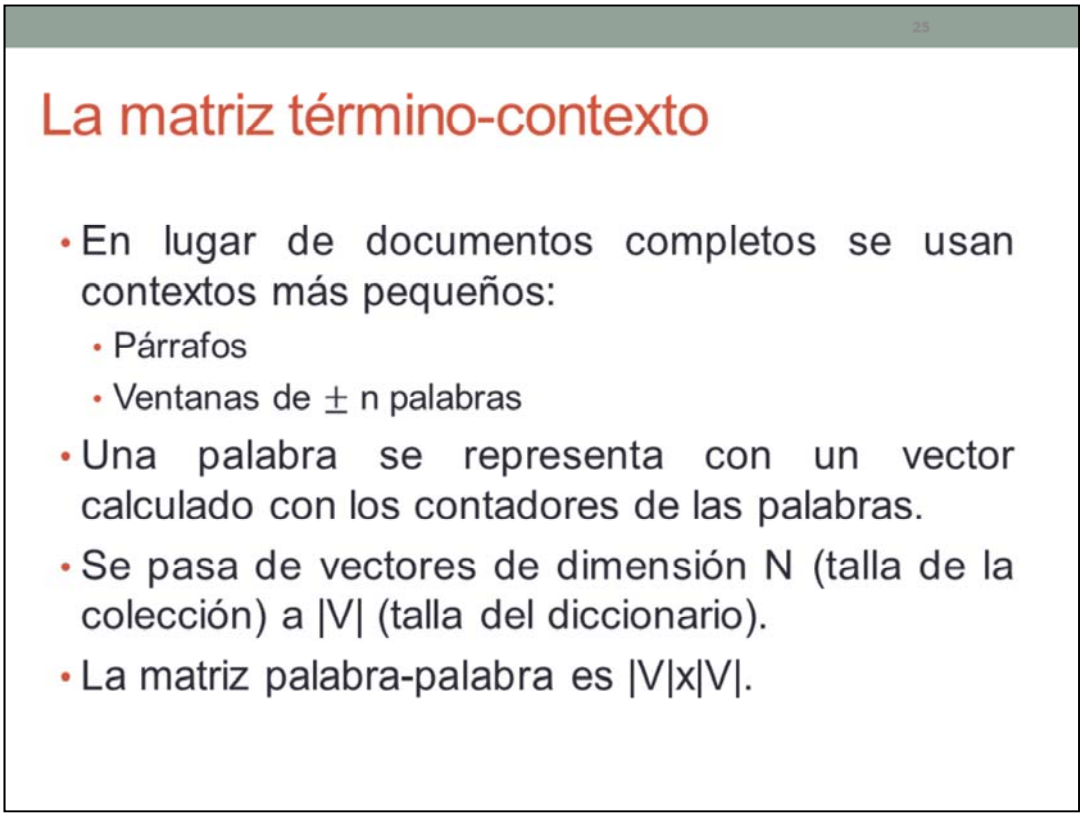
De la misma forma podemos asociar vectores a términos.

Las palabras en la matriz término-documento

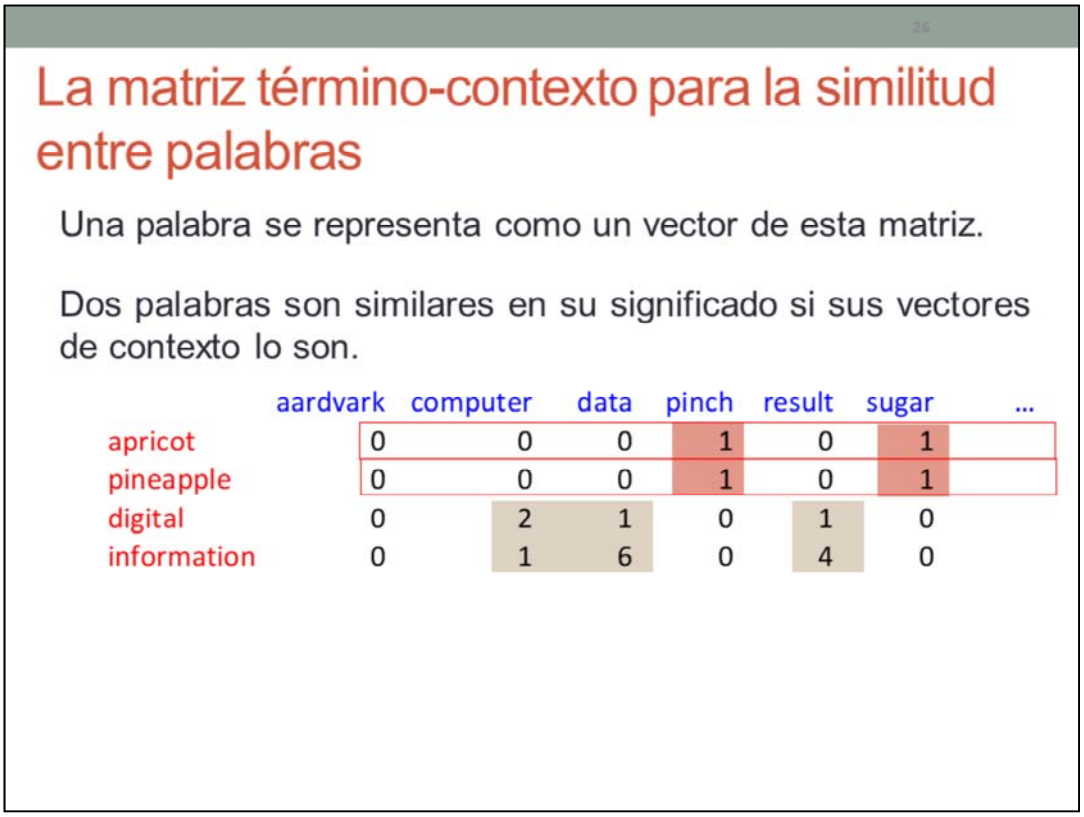
Dos palabras son similares si sus vectores lo son.

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Y calcular similitudes de términos en base a las similitudes de sus vectores.



Ahora definimos una matriz palabra-palabra en la que dada una palabra objetivo, se recogen estadísticas de ocurrencias de otras palabras en los contextos de la palabra objetivo observados en un corpus de entrenamiento (una serie de documentos). En esta aproximación la talla de los vectores de representación es la talla del vocabulario.

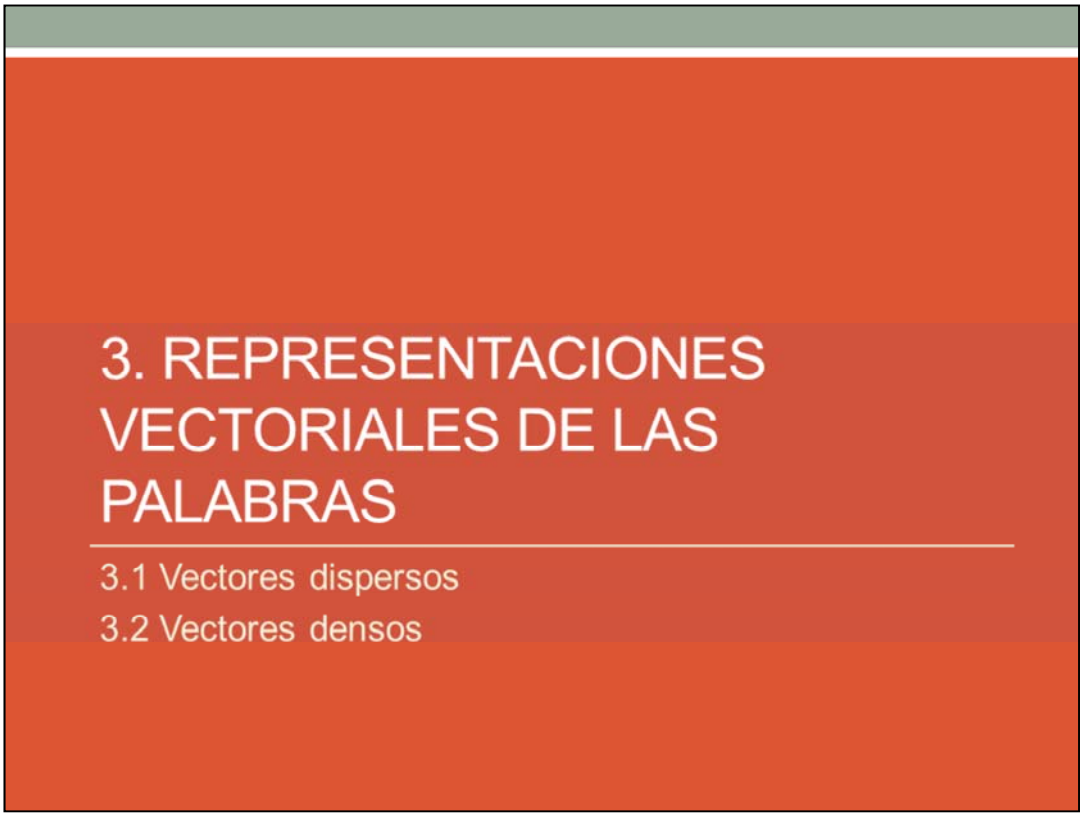


La representación vectorial de las palabras con vectores dispersos requiere la construcción de una matriz que recoge para cada palabra del vocabulario V las ocurrencias de las diferentes palabras del vocabulario en una ventana de cierta longitud alrededor de la palabra objetivo en un corpus de entrenamiento.

La matriz término-contexto

- Se muestra únicamente una pequeña parte de la matriz palabra-palabra: 4x6, de 50,000 x 50,000
  - Es muy dispersa
  - La mayor parte de los componentes son 0.
- La talla de la ventana depende del objetivo:
  - Tallas menores captan representaciones sintácticas (1-3 muy sintácticas)
  - Tallas mayores captan representaciones semánticas (4-10 más semánticas)

Este modelo que solamente cuenta con la frecuencia de ocurrencia en el contexto ha sido mejorado posteriormente.



Vamos a presentar dos aproximaciones para las representaciones vectoriales de las palabras: una que usa vectores dispersos y otra que usa vectores densos.

29

Representaciones vectoriales de las palabras

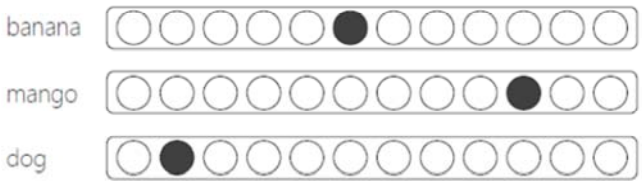
Representación one-hot (representación local):

Cada término de un vocabulario V es representado como un vector binario:

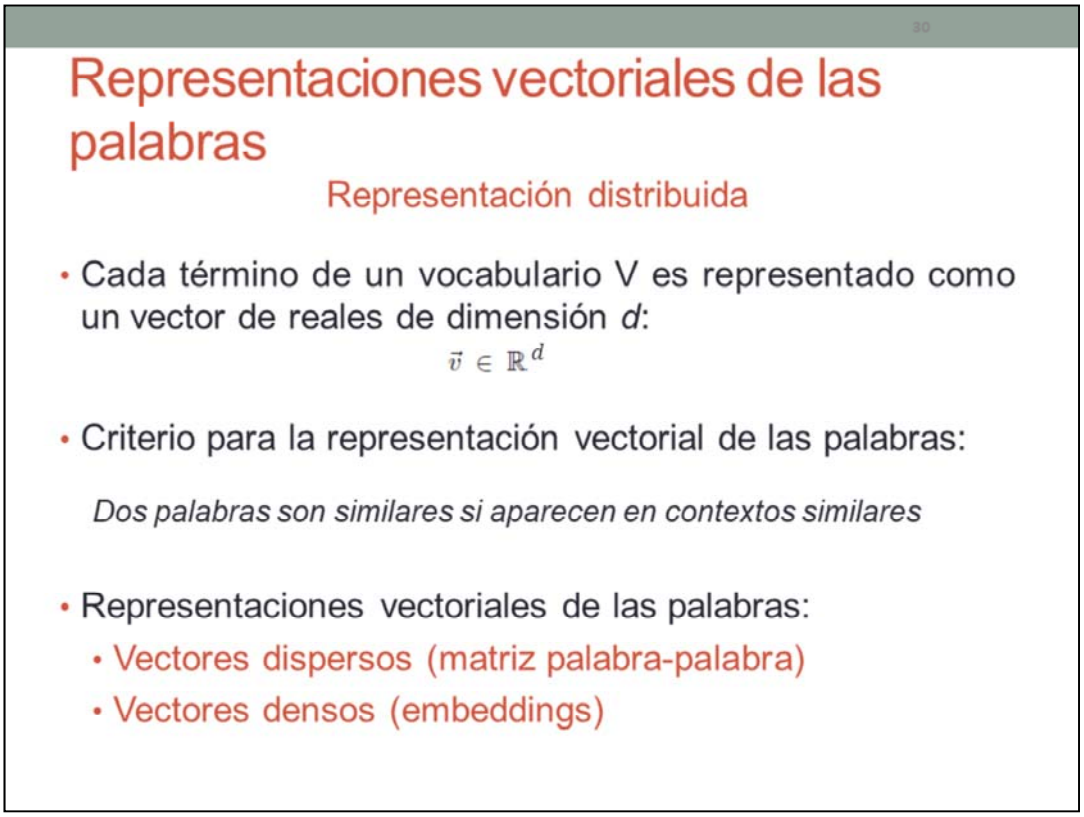
$$\vec{v} \in \{0,1\}^{|V|}$$

Donde uno de los valores del vector es 1 y el resto 0.

Cada posición en el vector tiene asociado un término.



La representación one-hot es muy simple, usa un vector binario.



Llamamos representación distribuida para distinguirla de la representación one-hot.

Se caracteriza por usar vectores de números reales y que trata de representar la semántica de las palabras.

3.1. Vectores dispersos

- Las simples frecuencias de ocurrencia en unos textos de referencia no representan una buena medida para la asociación de palabras.
- Se necesita una medida que represente mejor si una palabra de contexto es particularmente informativa sobre la palabra objetivo:

  Positive Pointwise Mutual Information (PPMI)

3.1. Vectores dispersos

Positive Pointwise Mutual Information (PPMI)

Una medida de cuánto más coocurren dos palabras comparado con lo esperado si fueran independientes

$$\text{PMI}(X, Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

PMI entre dos palabras:

$$\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Es una medida muy útil cuando se requiere encontrar palabras que están fuertemente relacionadas.

Vectores densos versus dispersos

¿Por qué vectores densos?

- Vectores de menor dimensión pueden ser usados más fácilmente como características en herramientas de machine learning (menos parámetros a ajustar)
- Son capaces de generalizar mejor que los contadores explícitos.
- Suelen captar mejor la sinonímia:

Vectores densos versus dispersos

- Los vectores PPMI son:
  - ✓grandes (longitudes 20,000-50,000)
  - ✓dispersos (la mayor parte de los componentes son cero)
- Alternativa: estimar vectores que sean:
  - ✓pequeños (longitudes 200-1000)
  - ✓densos (la mayor parte de los componentes son distintos de cero)

3.2. Vectores densos

Tres métodos para obtener vectores densos:

- Singular Value Decomposition (SVD)

  un caso particular es el conocido Latent Semantic Analysis (LSA)

- Neural Language Model

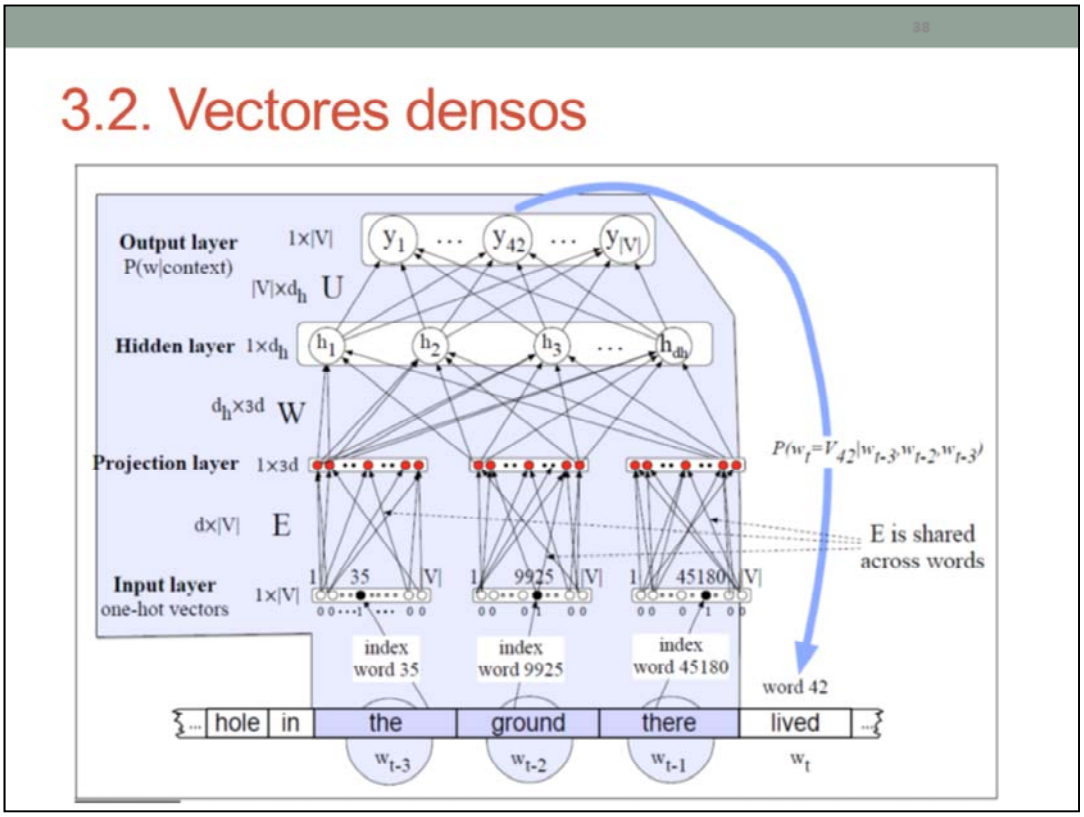
  basado en modelos predictivos skip-grams and CBOW

- Brown clustering

3.2. Vectores densos: word2vec

- **Skip-gram** (Mikolov et al. 2013a) **CBOW** (Mikolov et al. 2013b)
- Se aprenden embeddings como parte del proceso de predicción de palabras.
- Se entrena una red neuronal para la predicción de palabras vecinas
  - Inspirado en modelos de lenguaje neuronales.
  - En el proceso, se aprenden embeddings densos para las palabras del corpus de entrenamiento.

Este es un método para la estimación de vectores densos, basado en el concepto de modelo de lenguaje.



En la figura se muestra un esquema de una red feedforward para representar secuencias de palabras.

La entrada es una secuencia con las representaciones de las 3 palabras anteriores (la historia pasada para el tiempo t), esta entrada es procesada para obtener una representación vectorial más compacta (basada en embeddings). La red cuenta con una capa oculta y la capa de salida proporciona una probabilidad para cada una de las palabras del vocabulario.

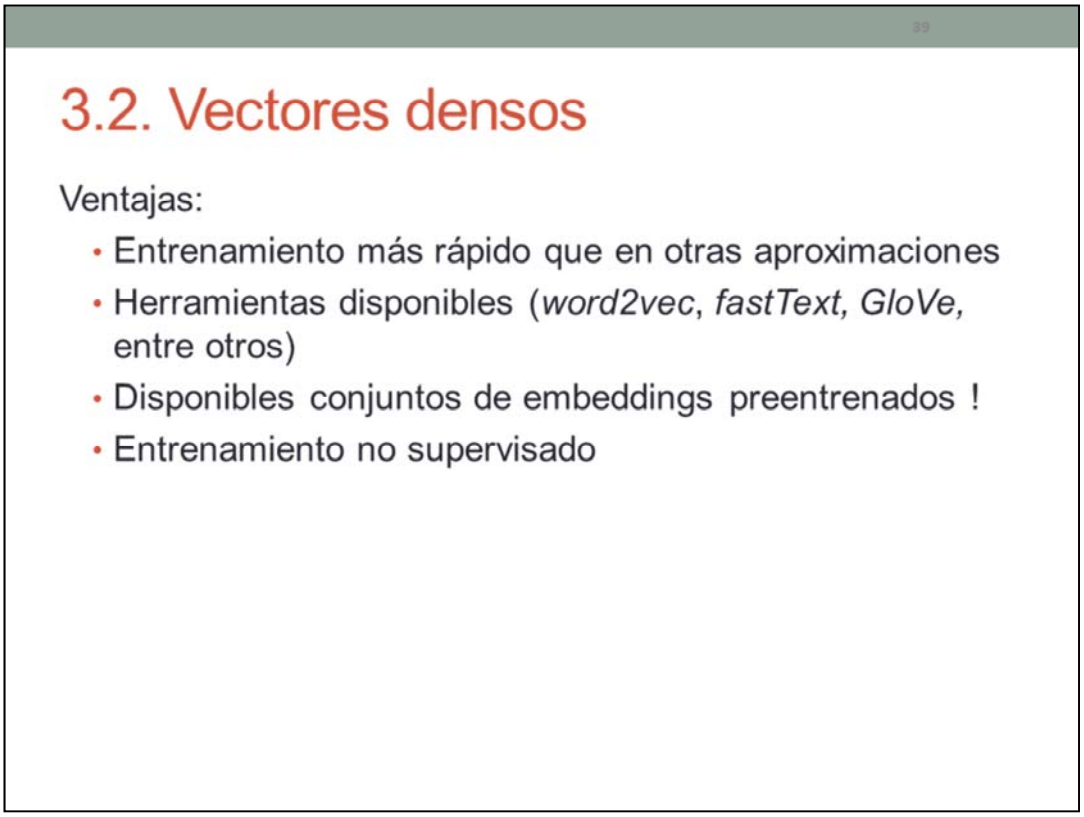
Partimos de un vocabulario V y de un diccionario (matriz) de embeddings E que proporciona la representación vectorial de dimensión d para cada palabra de V.

En la figura de asume una ventana en el texto de longitud 3, de forma que la entrada a la red es la representación inicial de las 3 palabras del contexto.

Esta representación inicial es un vector one-hot para cada palabra (un vector de dimensión la talla de V en el cual solamente un valor es 1 y el resto 0).

Dadas las tres palabras anteriores, buscamos sus índices, creamos 3 vectores one-hot y luego multiplicamos cada uno por la matriz de embeddings E.

Esta entrada es proporcionada a una red feedforward cuya salida es una softmax con una distribución de probabilidad sobre el vocabulario V.



Actualmente están disponibles embeddings preentrenados con grandes corpus de datos para una gran diversidad de lenguas.

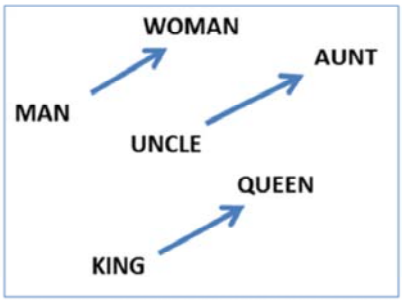
Actualmente se ha introducido una evolución de este tipo de embeddings que contempla una representación densa dependiente del contexto.

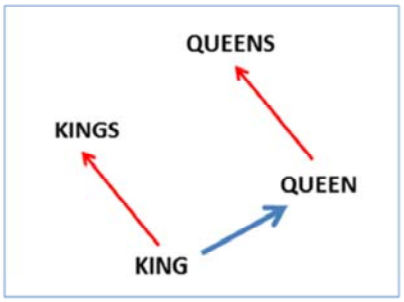
3.2. Vectores densos: word2vec

Propiedades: capturan relaciones semánticas

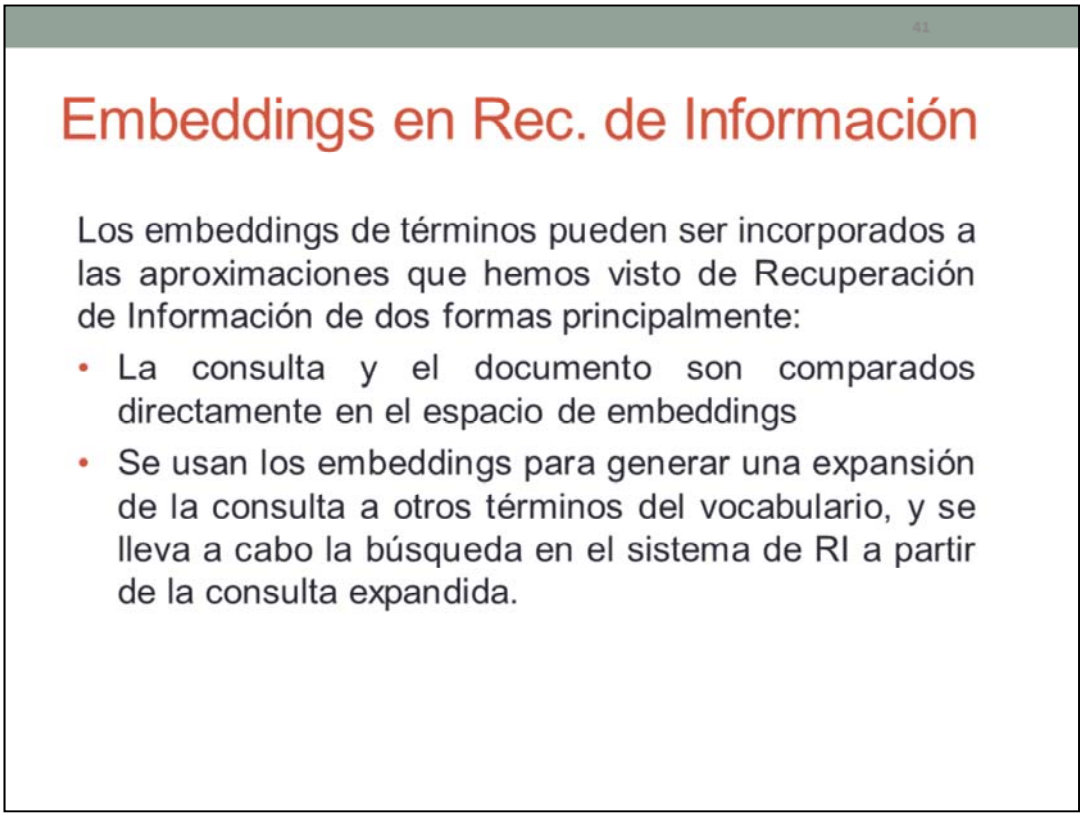
vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')

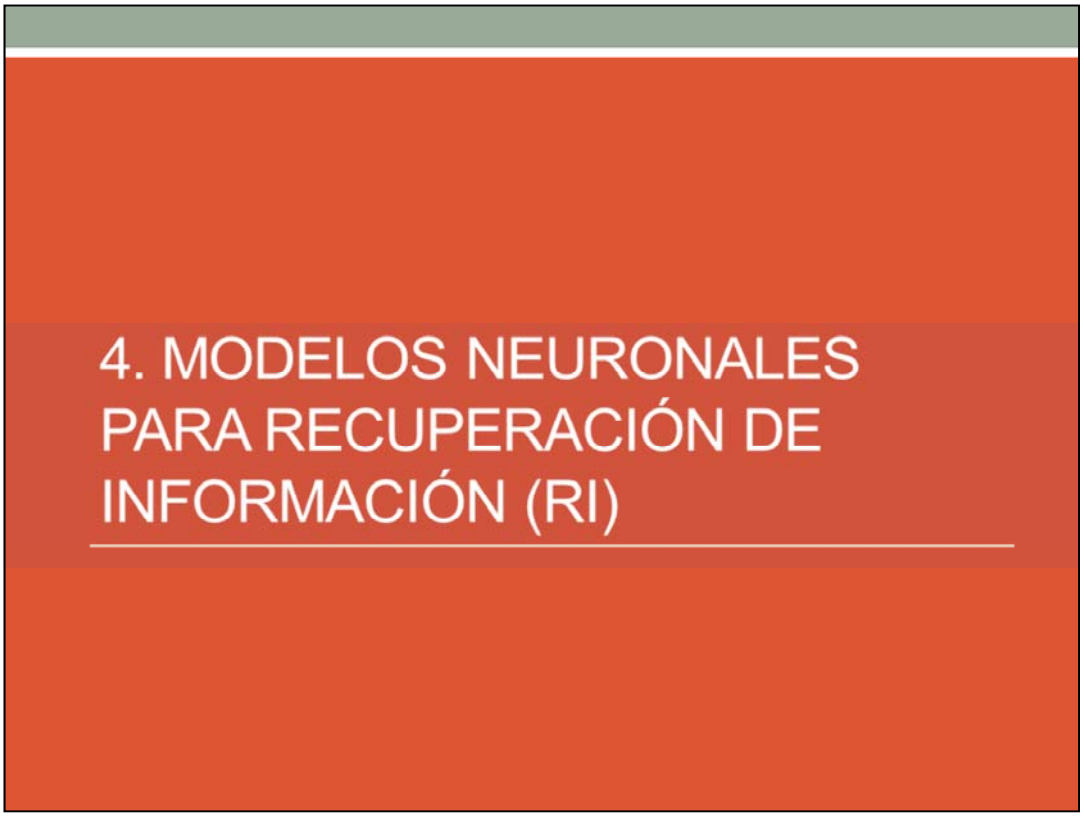




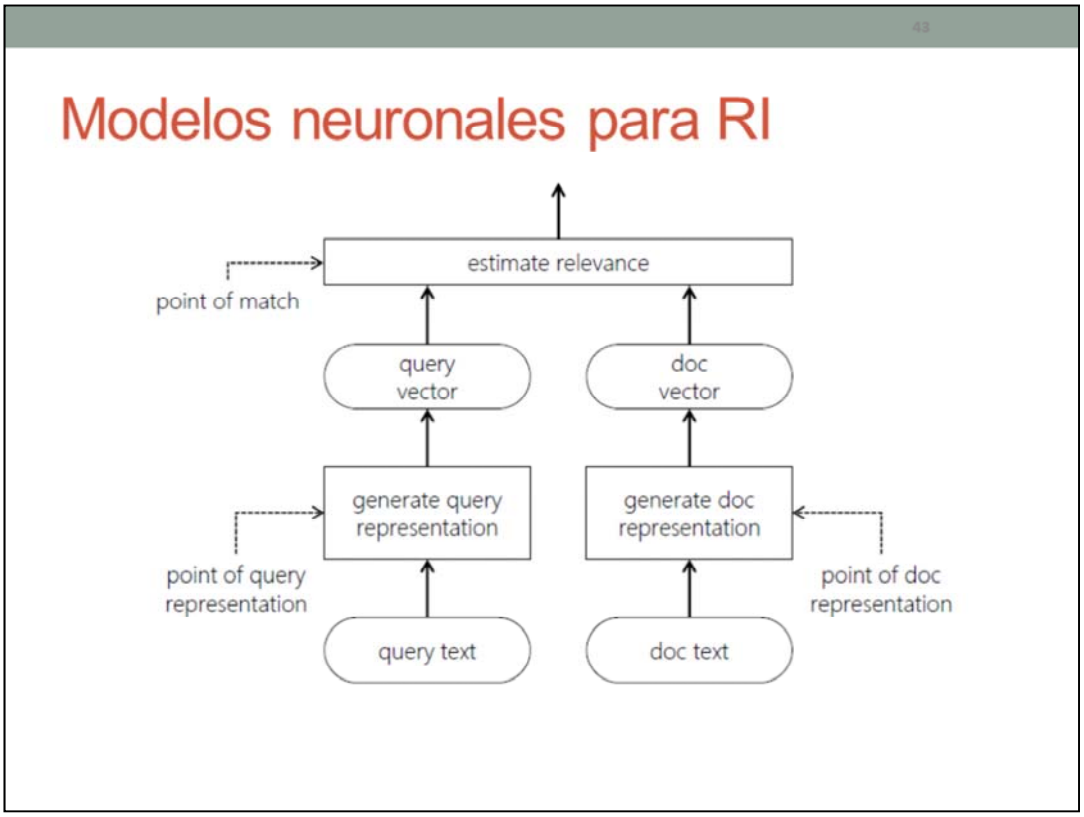
Se han presentado trabajos de similitud semántica en los que se muestra cómo la representación basada en embeddings es capaz de capturar relaciones semánticas entre palabras de una cierta complejidad, tal como se ilustra en la dispositiva.



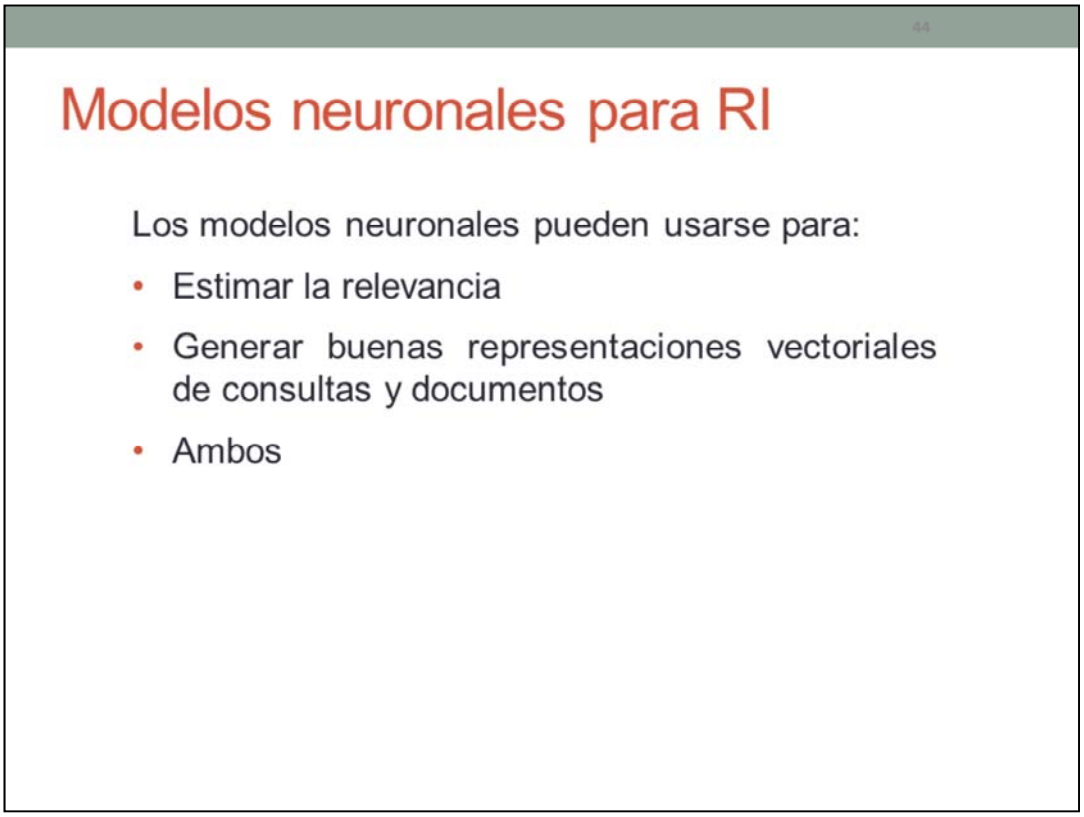
La mayoría de los métodos neuronales existentes para RI se centran en la coincidencia inexacta usando embeddings de términos. Estos enfoques pueden clasificarse ampliamente como aquellos que comparan la consulta con el documento directamente en el espacio de embeddings, y aquellos que utilizan embeddings para generar candidatos de expansión de consultas adecuados a partir de un vocabulario global y luego realizar la recuperación en función de la consulta ampliada.



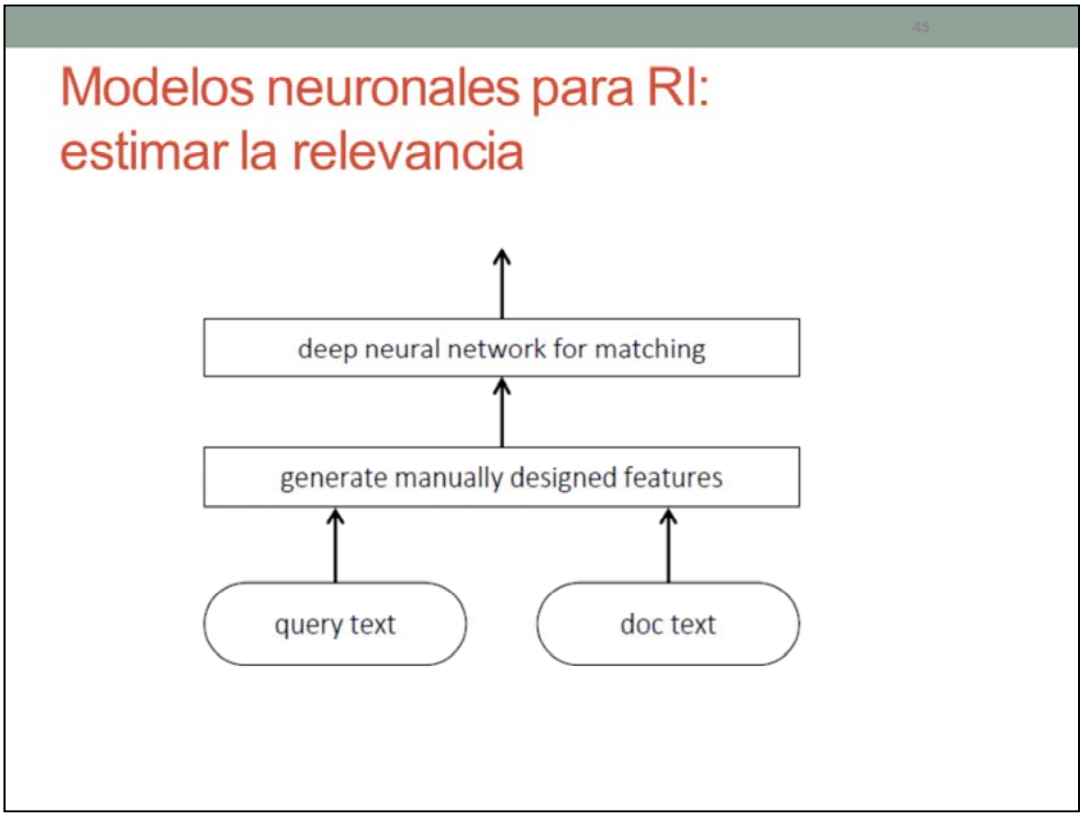
Los modelos neuronales para RI usan redes neuronales para ranquear los resultados de la consulta.



El esquema representa los diferentes puntos de la arquitectura del sistema de RI en los que podrían actuar las redes neuronales.

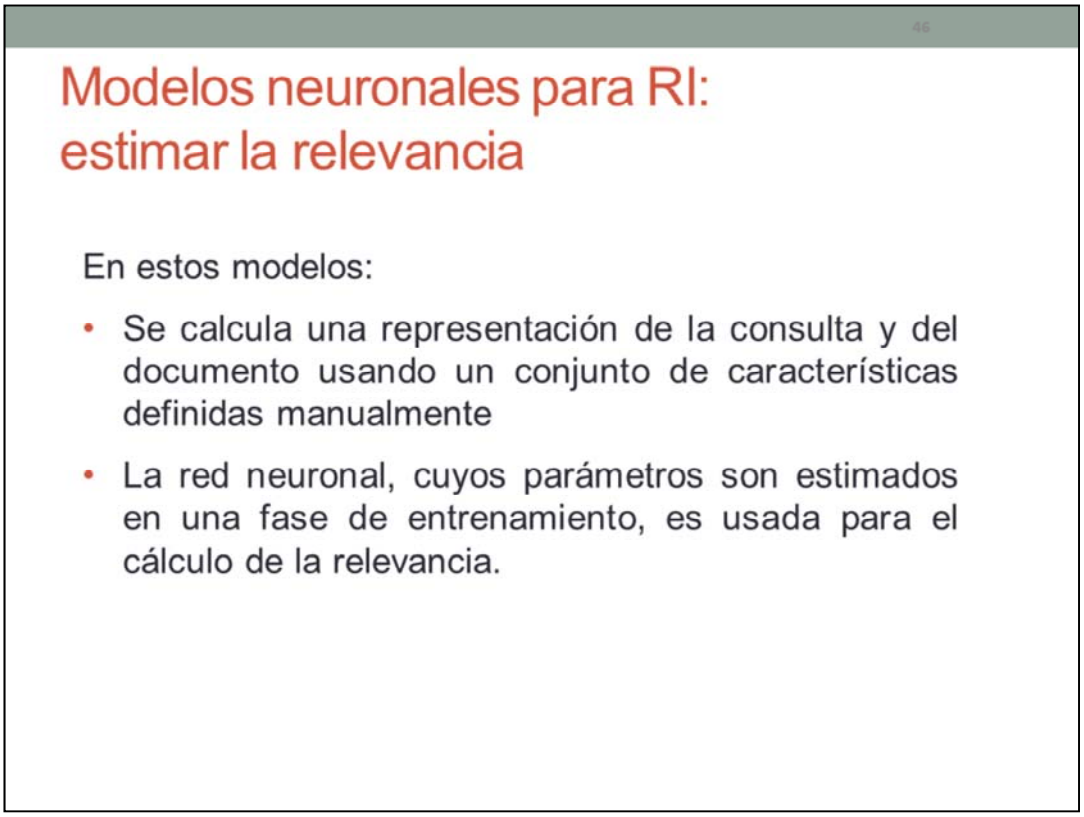


La RI comprende realizar tres pasos principales: generar una representación de la consulta que especifique la necesidad de información, generar una representación del documento que capture la distribución sobre la información contenida y hacer coincidir la consulta y las representaciones del documento para estimar su relevancia mutua. Todos los enfoques neuronales existentes para la RI se pueden categorizar en función de si influyen en la representación de la consulta, la representación del documento o en la estimación de relevancia.



Uso de una red neuronal para la estimación de la relevancia del documento para la consulta.

Uno de los modelos de machine learning dentro de la aproximación Learning to Rank (L2R)

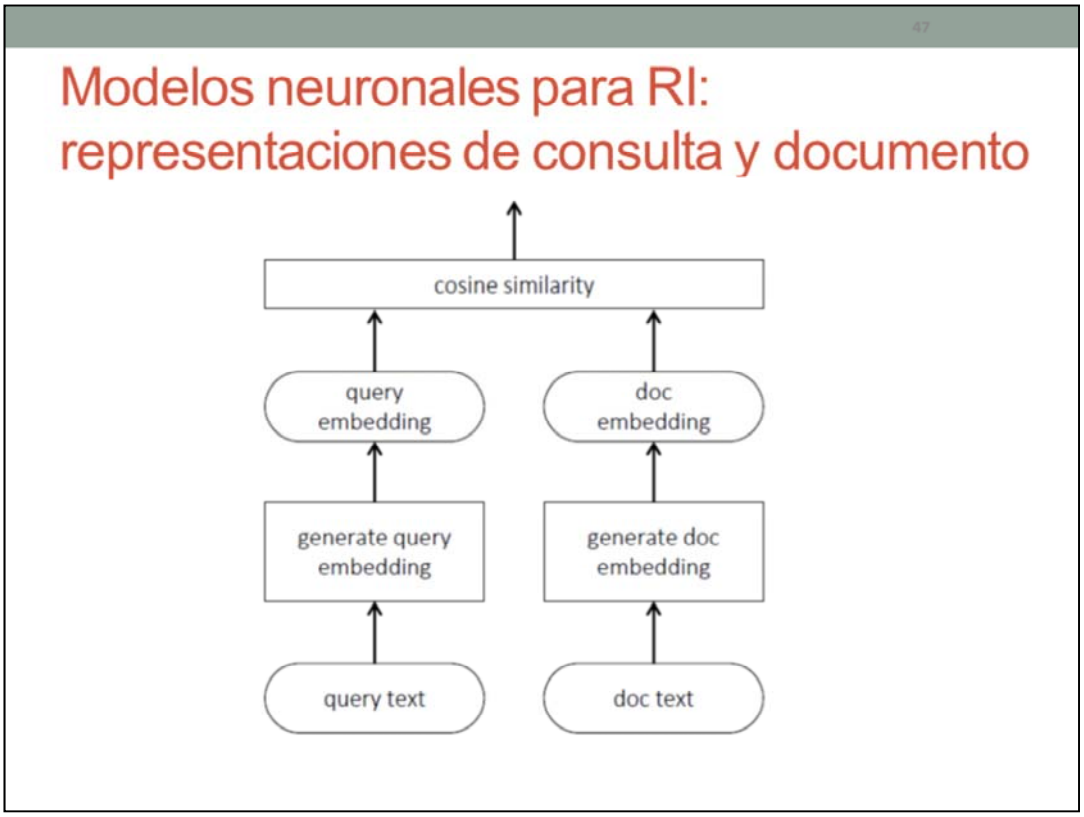


Cuando la red neuronal se usa para el cálculo de la relevancia, hace falta un proceso de entrenamiento de la red.

El conjunto de datos de entrenamiento para el modelo consta de un conjunto de consultas y un conjunto de documentos por consulta.

Dado un par consulta-documento representado mediante un vector de características (números reales generalmente), se entrena un modelo que asigna al vector de características una puntuación (generalmente un valor real).

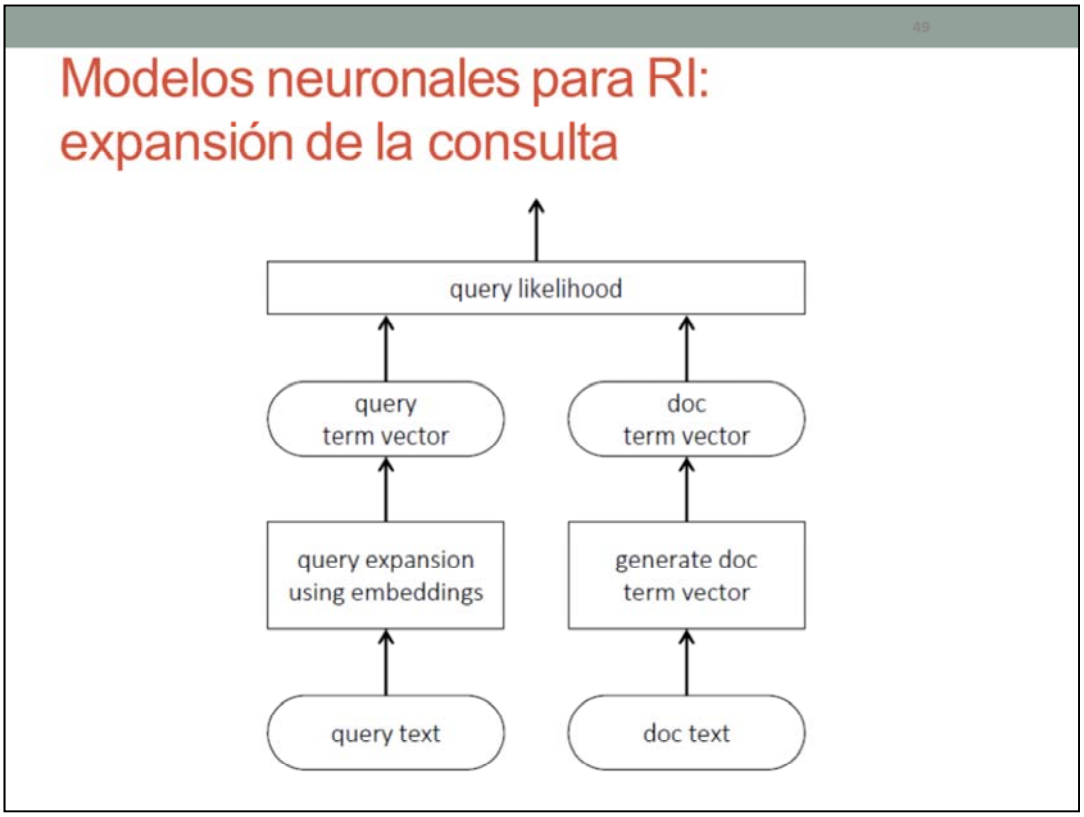
A lo largo de los años, se han empleado diferentes modelos de machine learning para aprender esta tarea: las máquinas de vectores de soporte, los árboles de decisión, y las redes neuronales.



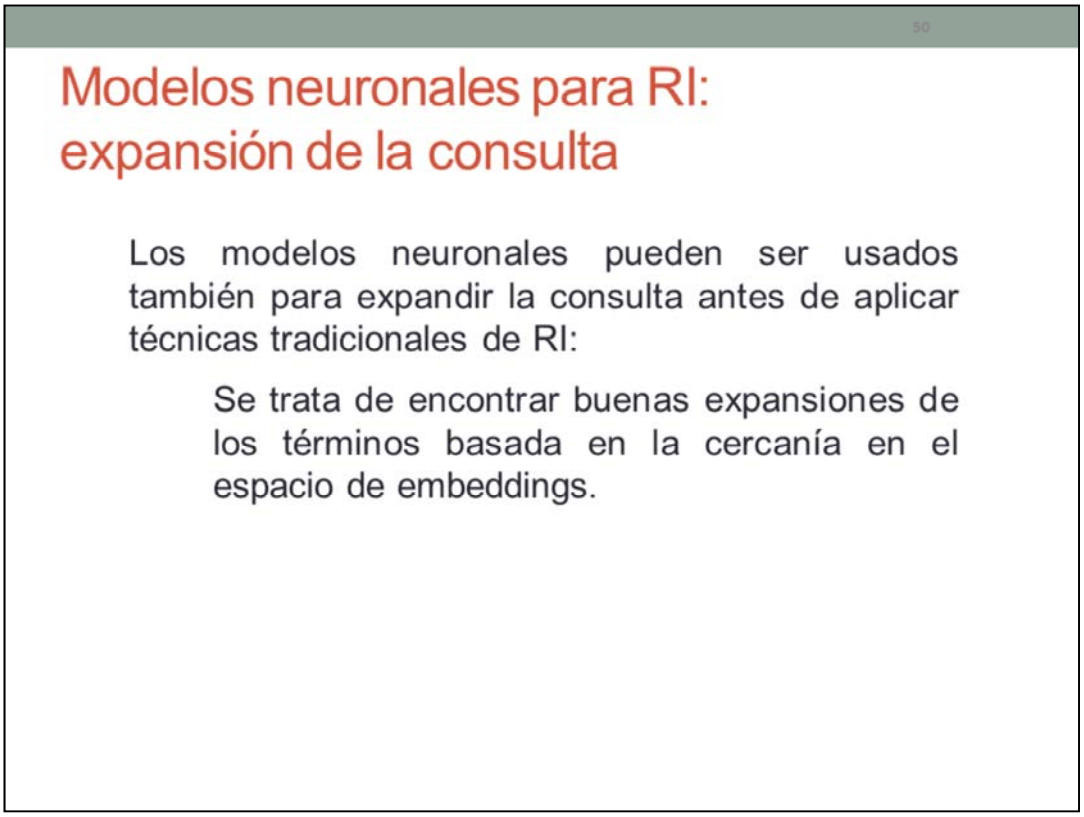
Uso de una red neuronal para la estimación de las representaciones vectoriales de la consulta y el documento.

Modelos neuronales para RI: representaciones de consulta y documento

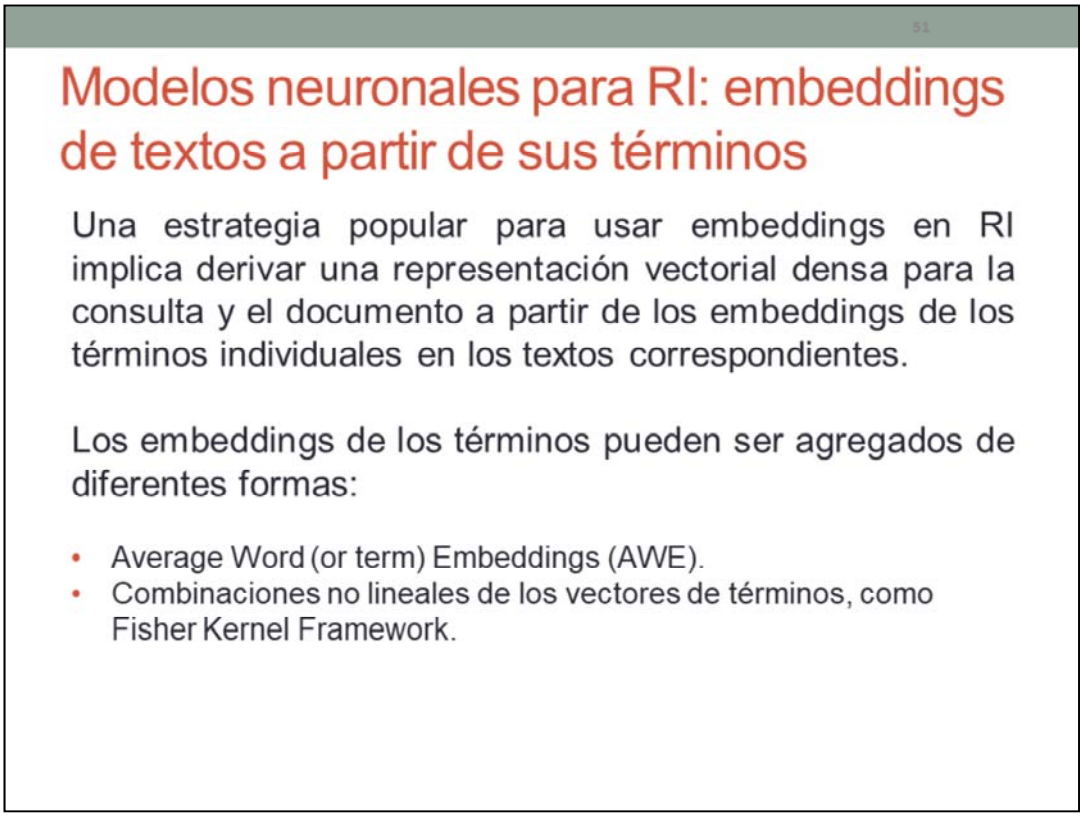
- Otros modelos neuronales para RI participan en la estimación de las representaciones vectoriales (embeddings) del texto de la consulta y el documento.
- Se usan dentro de los modelos RI tradicionales con métricas de similitud simples (por ejemplo, similitud de coseno).
- Estos modelos pueden aprender embeddings especializados para tareas de RI, o embeddings genéricos de forma no supervisada en base a textos independientes.



Uso de una red neuronal para la expansión de la consulta.



Utilizan embeddings para generar candidatos de expansión de consultas adecuados a partir de un vocabulario global y luego realizar la recuperación en función de la consulta ampliada.



Cómo obtener representaciones vectoriales densas (embeddings) de los textos (consulta y documento) a partir de los embeddings de sus términos.