# TEMA 9. OTROS MODELOS EN RECUPERACIÓN DE INFORMACIÓN

# Contenidos

# 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*

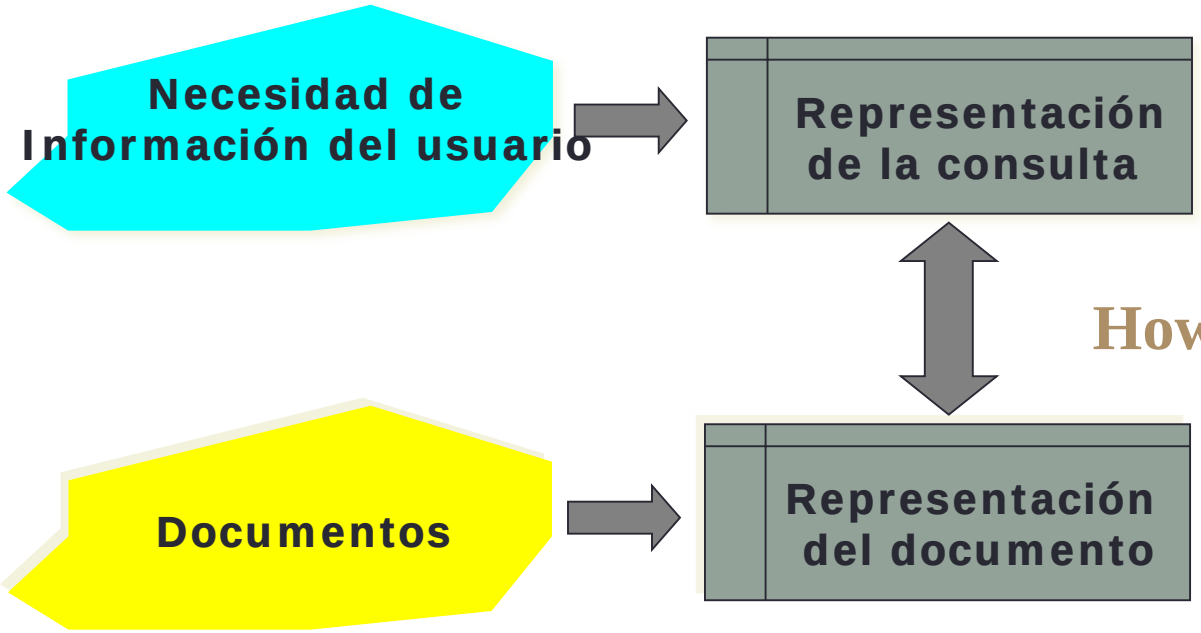
*Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017*

# 1. MODELO PROBABILÍSTICO

1.1. Introducción

1.2. Modelo probabilístico

# 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.

# 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).

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

*$f_i$ 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

# 2. REPRESENTACIÓN DE LAS PALABRAS

2.1. Las palabras y su contexto

2.2. Similitud entre palabras

# 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.

# 2.2 Similitud entre palabras

"**fast**" es similar a "**rapid**"

"**tall**" es similar a "**height**"

Question answering:

*Q: "How **tall** is Mt. Everest?"*
*Candidate A: "The official **height** of Mount Everest is 29029 feet"*

# 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 |

# Matriz término-documento

Dos documentos 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 |

# 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 |

# 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 |

# La matriz término-contexto

- En lugar de documentos completos se usan contextos más pequeños:
  - Párrafos
  - Ventanas de ± n palabras
- Una palabra se representa con un vector calculado con los contadores de las palabras.
- Se pasa de vectores de dimensión N (talla de la colección) a |V| (talla del diccionario).
- La matriz palabra-palabra es |V|x|V|.

# La matriz término-contexto para la similitud entre palabras

Una palabra se representa como un vector de esta matriz.

Dos palabras son similares en su significado si sus vectores de contexto lo son.

| | aardvark | computer | data | pinch | result | sugar | ... |
|---|---|---|---|---|---|---|---|
| apricot | 0 | 0 | 0 | 1 | 0 | 1 | |
| pineapple | 0 | 0 | 0 | 1 | 0 | 1 | |
| digital | 0 | 2 | 1 | 0 | 1 | 0 | |
| information | 0 | 1 | 6 | 0 | 4 | 0 | |

# 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)

# 3. REPRESENTACIONES VECTORIALES DE LAS PALABRAS

3.1 Vectores dispersos

3.2 Vectores densos

# Representaciones vectoriales de las palabras

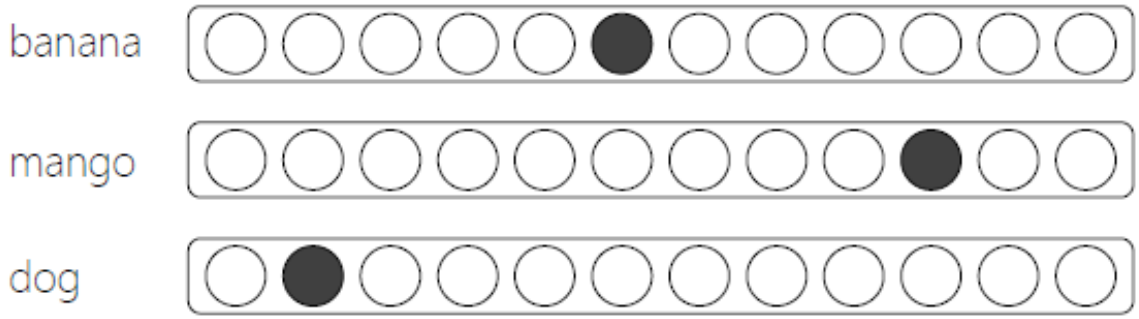
Representación one-hot (representación local):

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.



# Representaciones vectoriales de las palabras

## Representación distribuida

- Cada término de un vocabulario V es representado como un vector de reales de dimensión *d*:

$$\vec{v} \in \mathbb{R}^d$$

- Criterio para la representación vectorial de las palabras:

*Dos palabras son similares si aparecen en contextos similares*

- Representaciones vectoriales de las palabras:
  - Vectores dispersos (matriz palabra-palabra)
  - Vectores densos (embeddings)

# 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

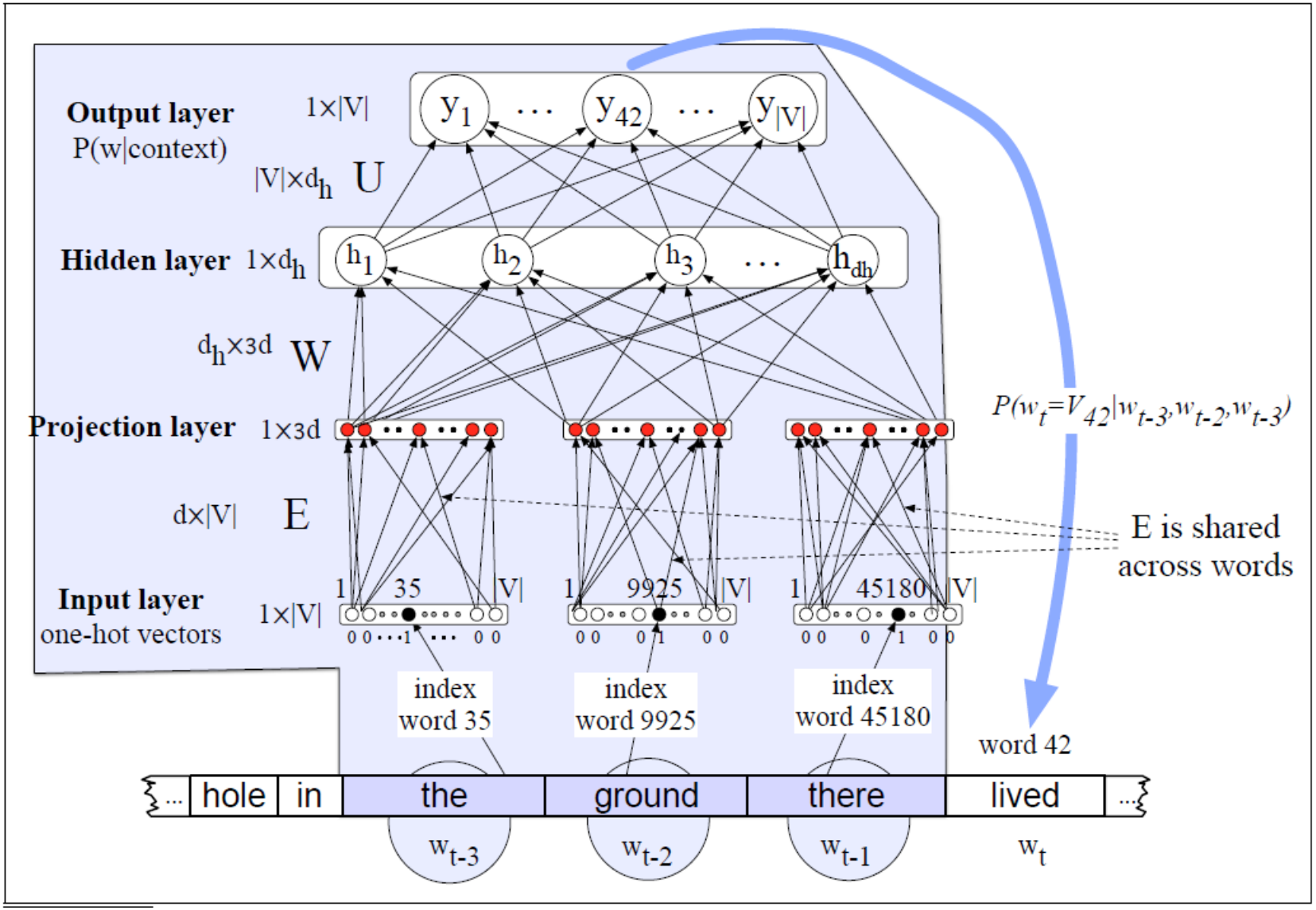
  basado en modelos predictivos skip-grams and CBOW

- 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.

# 3.2. Vectores densos



# 3.2. Vectores densos

Ventajas:

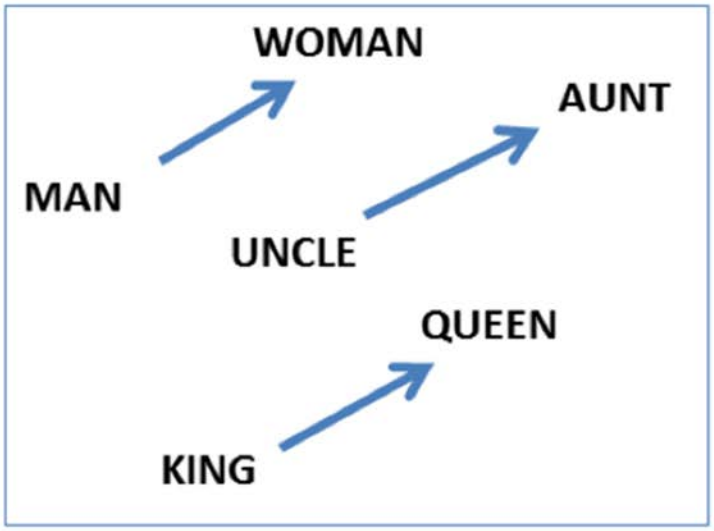
- Entrenamiento más rápido que en otras aproximaciones
- Herramientas disponibles (*word2vec*, *fastText, GloVe,* entre otros)
- Disponibles conjuntos de embeddings preentrenados !
- Entrenamiento no supervisado

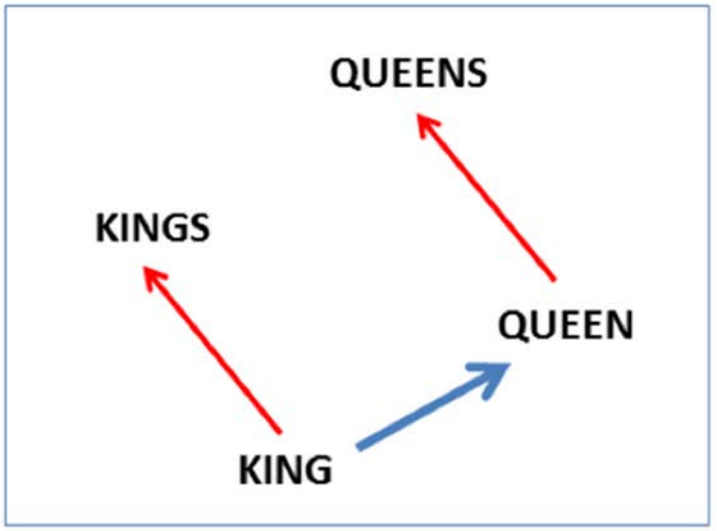
# 3.2. Vectores densos: word2vec

Propiedades: capturan relaciones semánticas

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

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



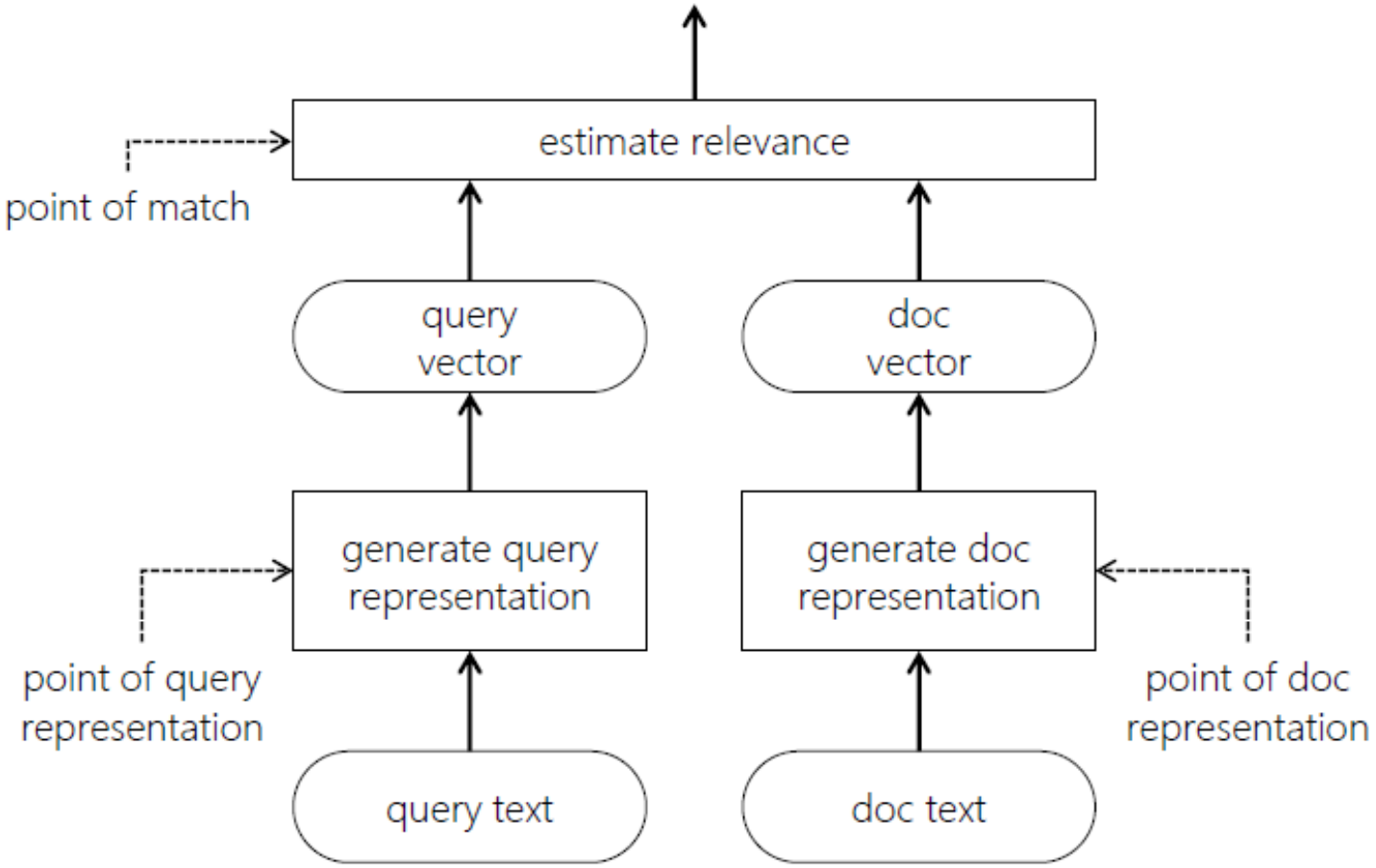
# Embeddings en Rec. de Información

Los embeddings de términos pueden ser incorporados a las aproximaciones que hemos visto de Recuperación de Información de dos formas principalmente:

- La consulta y el documento son comparados directamente en el espacio de embeddings
- Se usan los embeddings para generar una expansión de la consulta a otros términos del vocabulario, y se lleva a cabo la búsqueda en el sistema de RI a partir de la consulta expandida.

# 4. MODELOS NEURONALES PARA RECUPERACIÓN DE INFORMACIÓN (RI)

# Modelos neuronales para RI

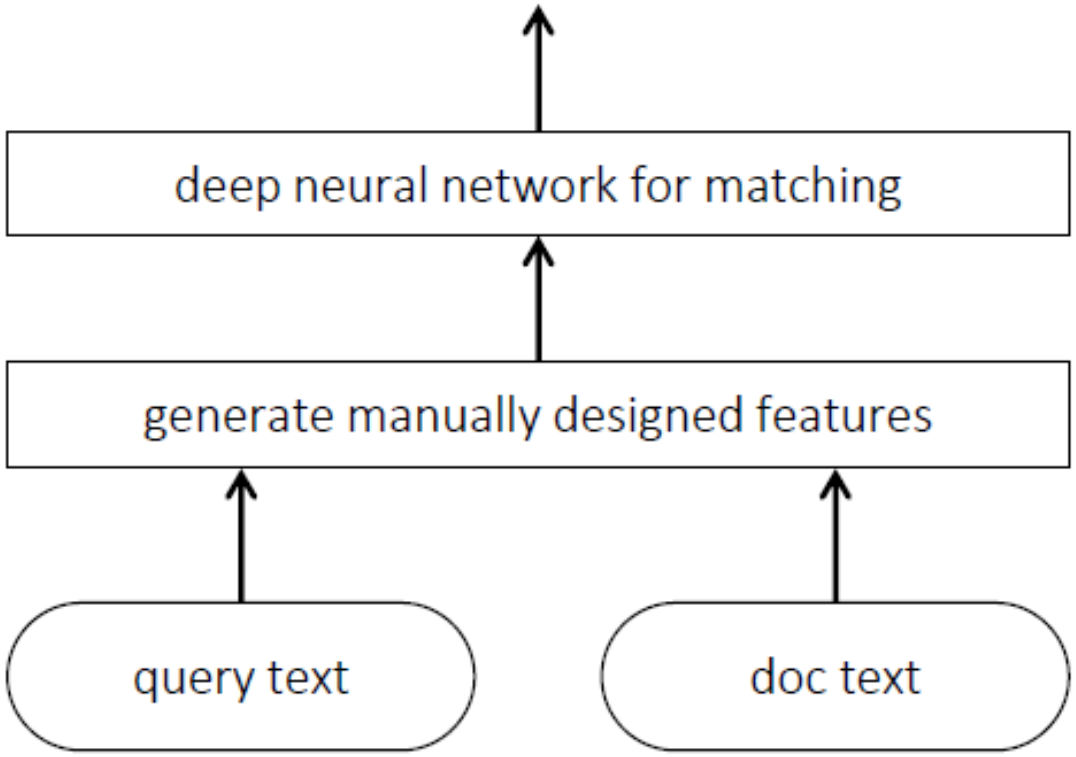


# Modelos neuronales para RI

Los modelos neuronales pueden usarse para:

- Estimar la relevancia
- Generar buenas representaciones vectoriales de consultas y documentos
- Ambos

# Modelos neuronales para RI: estimar la relevancia

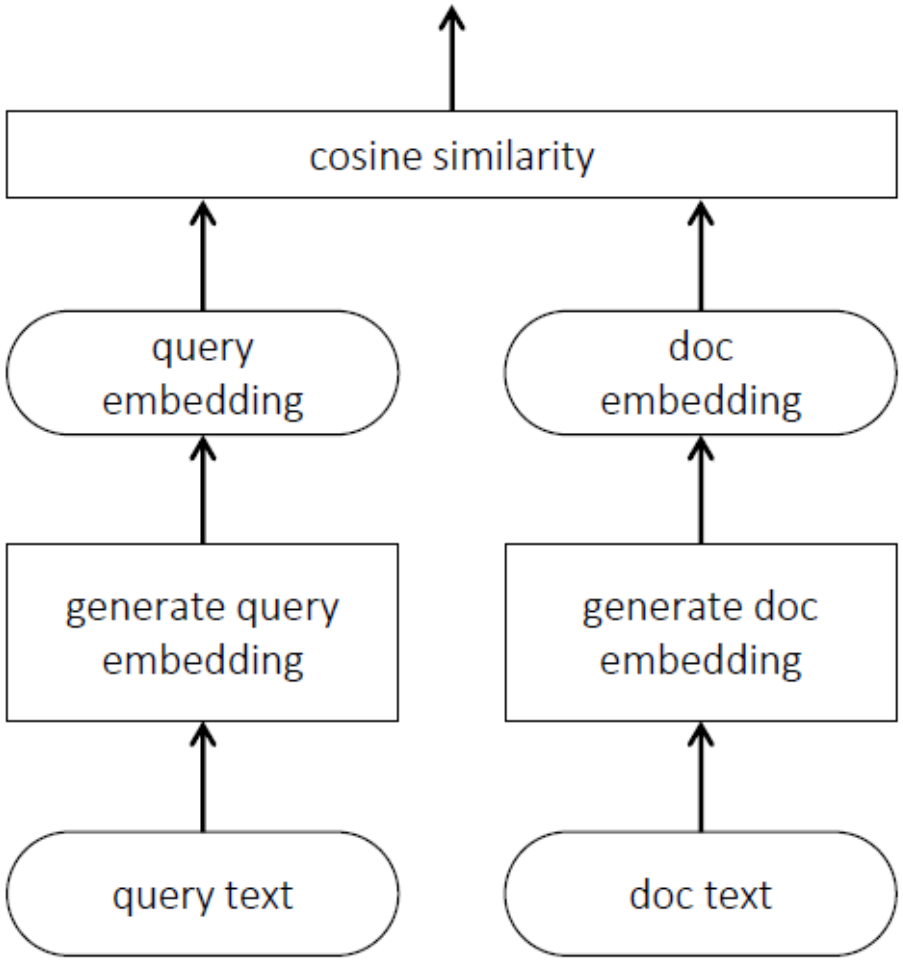


# Modelos neuronales para RI: estimar la relevancia

En estos modelos:

- Se calcula una representación de la consulta y del documento usando un conjunto de características definidas manualmente
- La red neuronal, cuyos parámetros son estimados en una fase de entrenamiento, es usada para el cálculo de la relevancia.

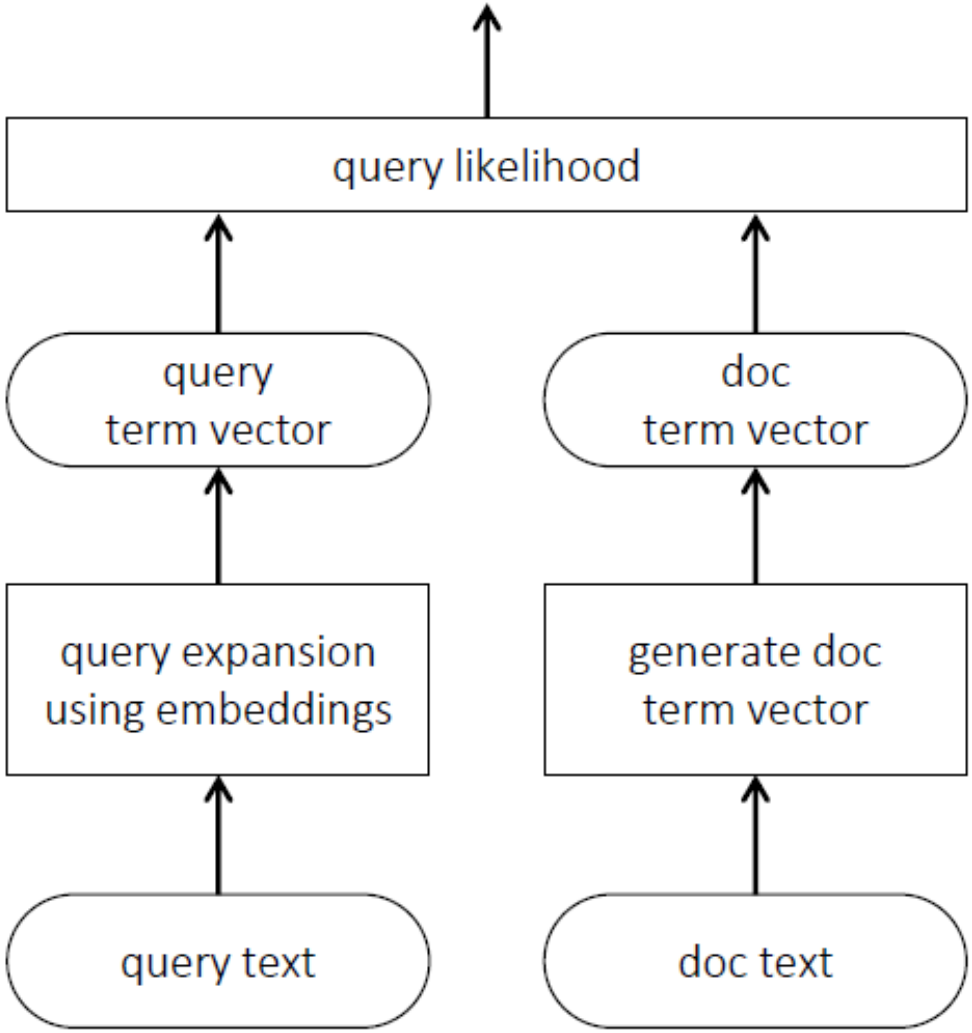
# Modelos neuronales para RI: representaciones de consulta y documento



# 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.

# Modelos neuronales para RI: expansión de la consulta



# Modelos neuronales para RI: expansión de la consulta

Los modelos neuronales pueden ser usados también para expandir la consulta antes de aplicar técnicas tradicionales de RI:

Se trata de encontrar buenas expansiones de los términos basada en la cercanía en el espacio de embeddings.

# Modelos neuronales para RI: embeddings de textos a partir de sus términos

Una estrategia popular para usar embeddings en RI implica derivar una representación vectorial densa para la consulta y el documento a partir de los embeddings de los términos individuales en los textos correspondientes.

Los embeddings de los términos pueden ser agregados de diferentes formas:

- Average Word (or term) Embeddings (AWE).
- Combinaciones no lineales de los vectores de términos, como Fisher Kernel Framework.