Recuperación Ranqueada

Rubén Francisco Manrique
rf.manrique@uniandes.edu.co

Modelos de recuperación ranqueada

- En lugar de un conjunto de documentos que satisfacen una expresión de consulta, en la recuperación ranqueada, ***el sistema devuelve un orden para los documentos (top) recuperados de la colección***.

  - El tamaño del set resultante no es un problema
  - Se muestran los top k resultados.
  - No se sobrecarga a el usuario con resultados

- ***Consultas de texto libre***: en lugar de un lenguaje de consulta de operadores y expresiones, la consulta del usuario es solo una o más palabras en un lenguaje humano.

Modelos de recuperación ranqueada

- En lugar de un conjunto de documentos que satisfacen una expresión de consulta, en la recuperación ranqueada, ***el sistema devuelve un orden para los documentos (top) recuperados de la colección***.

  - El tamaño del set resultante no es un problema
  - Se muestran los top k resultados.
  - No se sobrecarga a el usuario con resultados

- ***Consultas de texto libre***: en lugar de un lenguaje de consulta de operadores y expresiones, la consulta del usuario es solo una o más palabras en un lenguaje humano.

Todo se resumen a un puntaje de similitud…

Ranquear de acuerdo a un score de similitud entre la consulta y los documentos.

¿Pero como calcular esa similitud?

Empecemos con algo básico: Coeficiente de Jaccard

- Sobrelapamiento de términos como indicativo de similitud.
  - jaccard(A,B) = |A ∩ B| / |A ∪ B|
  - jaccard(A,A) = 1
  - jaccard(A,B) = 0 if A ∩ B = 0

- A y B no tienen que ser del mismo tamaño.
- La disparidad en el número de términos en los documentos puede se un problema.
- Siempre asigna un score entre 0 y 1.

Problemas con el Coeficiente de Jaccard

- Necesitamos una forma más sofisticada de normalizar la longitud.
- No tiene en cuenta la frecuencia de los términos (cuántas veces aparece un término en un documento).
- Los términos raros en una colección son más informativos que los términos frecuentes. Jaccard no considera esta información – ***PODER DISCRIMINATIVO.***

Matrices termino-documento

- Considere el número de ocurrencias de un término en un documento:
  - Cada documento es un vector columna en $\mathbb{N}^V$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Modelo Bolsa de Palabras (Bag of Words) BOW

- La representación vectorial no considera el orden de las palabras en un documento.

- Las siguientes oraciones tienen la misma representación.
  - El perro es más rápido que el caballo.
  - El caballo es más rápido que el perro.

- Se denomina modelo BOW.

Modelo Bolsa de Palabras (Bag of Words) BOW

- La frecuencia del término t en el documento d se define como $tf_{t,d}$
  - Frecuencia = Contar

- Queremos usar $\boldsymbol{tf}$ al calcular las puntuaciones de coincidencia/similitud entre los documentos de la colección y la consulta. ¿Cómo lo podemos realizar?

- La frecuencia del término sin procesar no es lo que queremos:
  - Un documento con 10 ocurrencias del término es más relevante que un documento con 1 ocurrencia del término.
  - Pero no 10 veces más relevante.

- La relevancia no aumenta proporcionalmente con la frecuencia del término.

Modelo Bolsa de Palabras: Ponderación logarítmica

- La frecuencia log del termino t en el documento d es:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- $0 \rightarrow 0$, $1 \rightarrow 1$, $2 \rightarrow 1.3$, $10 \rightarrow 2$, $1000 \rightarrow 4$, etc.
- Puntué un documento para una consulta dada según la formula:

$$score = \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

- La puntuación es 0 si ninguno de los términos de consulta está presente en el documento.

- Nota: A veces se prefiere una normalización básica por el número total de términos.

Frecuencia documental

- La frecuencia documental del término t es el número de documentos en la colección donde t ocurre.
- Ejemplo suponga N=10000.

Término	Frecuencia Documental
agua	3997
intentar	8760

- ¿Qué palabra es para una mejor búsqueda (obtiene mayor peso)?

Los términos raros son mas informativos

- Los términos raros son más informativos que los términos frecuentes.
  - Recuerden las palabras de parada.

- Considere un término en la consulta que es raro en la colección (por ejemplo, *electroencefalografía*).

- Es muy probable que un documento que contenga este término sea relevante para la consulta *electroencefalografía*.

- Queremos un peso alto para términos raros en la ***colección*** como *electroencefalografía*. Mayor poder ***discriminatorio***.

Ponderación idf

- $\boldsymbol{df_t}$ es la frecuencia documental del termino $\boldsymbol{t}$ (número de documentos que contienen $\boldsymbol{t}$).
  - $\boldsymbol{df_t}$ es una medida inversa del grado de "informatividad" de t
  - $\boldsymbol{df_t \leq N}$

- Se define la ***frecuencia inversa documental*** del termino t como:

$$\mathrm{idf}_t = \log_{10}\left(\frac{N}{\mathrm{df}_t}\right)$$

- Se usa el logaritmo para amortiguar la división.

Ejemplo idf

- Suponga $N = 1.000.000$

termino	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

$$\mathrm{idf}_t = \log_{10}\left(\frac{N}{\mathrm{df}_t}\right)$$

Ponderación tf-idf

- $\boldsymbol{tf - id_t}$ pondera un término como el producto de su peso tf y su peso idf.

$$\mathrm{w}_{t,d} = tf.idf_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log_{10}\left(\frac{N}{\mathrm{df}_t}\right)$$

- Nótese:
  - tf información del término intra-documento.
  - idf información del término en la colección.

- Esquema de ponderación más conocido en recuperación de información.

¿Como puntuó una consulta dado un documento?

$$\text{Score}(q,d) = \sum_{t \in q \cap d} \text{tf.idf}_{t,d}$$

Documentos como vectores

Nuestra primera representación vectorial

Pasamos de una matriz binaria, a una de conteo y finalmente llegamos a una ponderada.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Cada documento ahora está representado por un vector de valor real de pesos tf-idf $\epsilon\ \mathbb{R}^{|V|}$

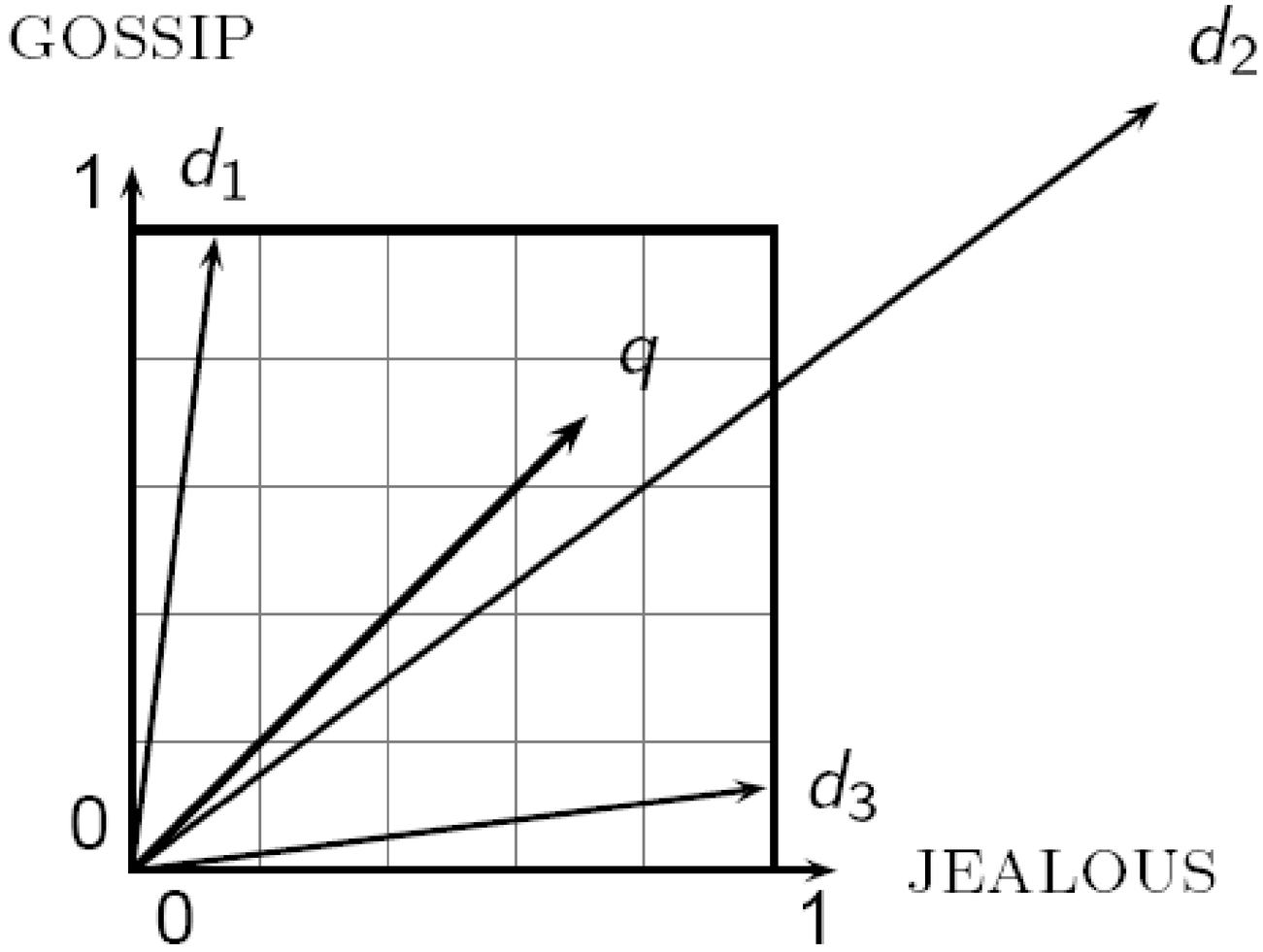
Documentos como vectores

- Tenemos un espacio |V|-dimensional.
- Los términos de nuestro vocabularios son los ejes en dicho espacio.
- Los documentos son puntos o vectores en este espacio.
- Espacio altamente dimensional: decenas de millones de dimensiones cuando aplica esto a un motor de búsqueda web.
- Estos son vectores muy dispersos: la mayoría de las entradas son cero.
- Idea clave: Si tengo documentos representados como vectores ranquéelos de acuerdo con su proximidad en este espacio vectorial.
  - Proximidad=Similaridad

Proximidad en términos de distancia?, mala idea

La distancia Euclidiana entre q y d_2 es mayor a las distancias con d_3 y d_1, a pesar que la distribución de palabras entre q y d_2 es muy similar.

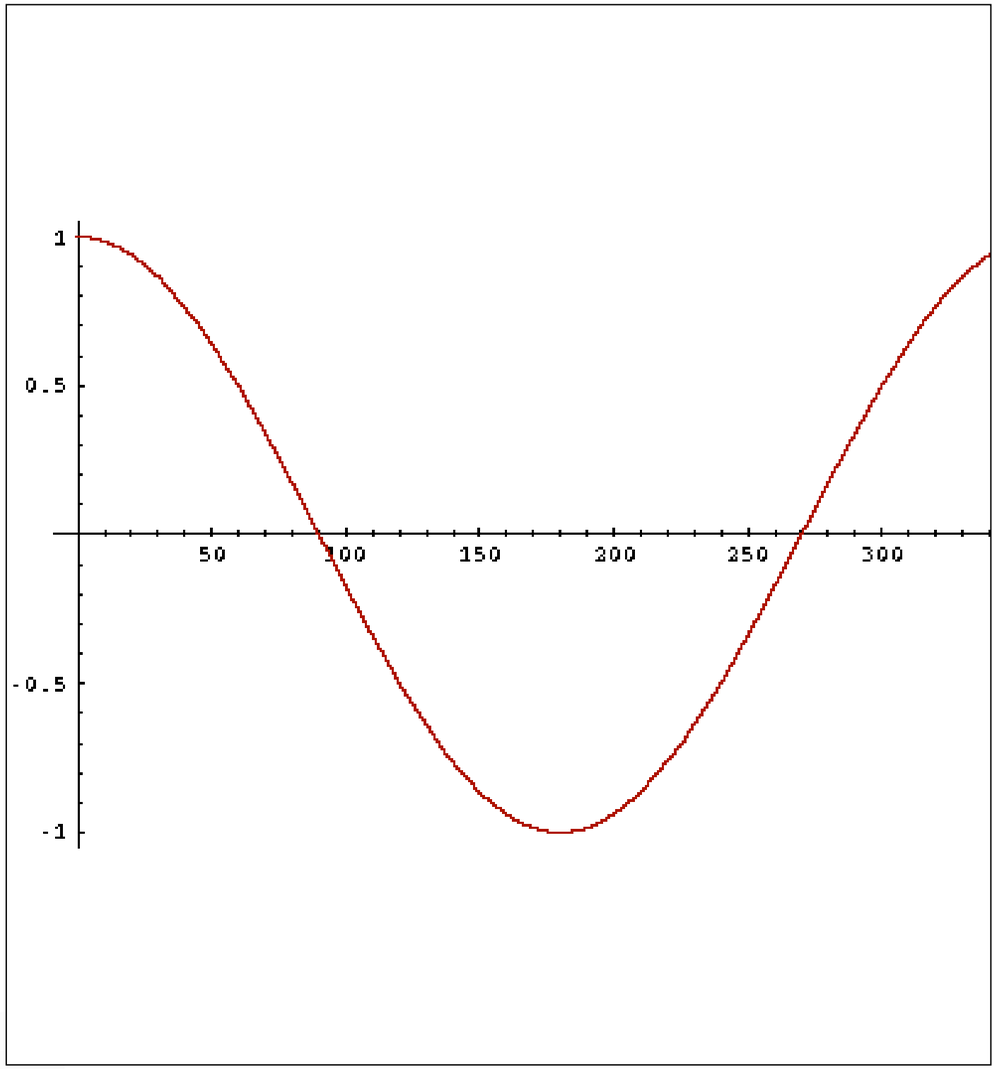
En aplicaciones de NLP es mejor ranquear documentos de acuerdo con el ángulo con la consulta.



De ángulos a cosenos

- Las dos nociones siguientes son equivalentes.
  - Clasifique los documentos en orden decreciente del ángulo entre la consulta y el documento
  - Clasifique los documentos en orden creciente de coseno (consulta, documento)

- El coseno es una función monótonamente decreciente para el intervalo [0, 180]



Norma L_2 y producto punto

- Un vector puede ser normalizado dividiendo cada uno de sus componente por la norma.

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$$

- Dividir un vector por su norma L_2 lo convierte en un vector unitario (de longitud) (en la superficie de la hiperesfera unitaria).

- El product punto de dos vectores por otro lado se define.

$$\vec{q} \cdot \vec{d} = \|\vec{q}\| \left\|\vec{d}\right\| \cos\theta$$

Coseno(consulta, documento)

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q} \bullet \vec{d}}{\|\vec{q}\|\|\vec{d}\|} = \frac{\vec{q}}{\|\vec{q}\|} \bullet \frac{\vec{d}}{\|\vec{d}\|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

- q_i es el peso del término i en la consulta
- d_i es el peso del término i en el documento

$\cos(\vec{q},\vec{d})$ es la ***similitud coseno*** entre $\vec{q}$ ***y*** $\vec{d}$

Ejemplo con tres documentos

3 documentos:

***CAS*: Cien años de soledad**

***CMA*: Crónica de una muerte anunciada**

***ATC*: El amor en los tiempos de colera.**

Término	CAS	CMA	ATC
pueblo	115	58	20
celos	10	7	11
altanero	2	0	6
colera	0	0	38

Frecuencia de los términos

Para simplificar este ejemplo, no se realizo ponderación idf.

Ejemplo con tres documentos

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

Término	CAS	CMA	ATC
pueblo	3.06	2.76	2.30
celos	2.00	1.85	2.04
altanero	1.30	0	1.78
colera	0	0	2.58

Normalización

Término	CAS	CMA	ATC
pueblo	0.789	0.832	0.524
celos	0.515	0.555	0.465
altanero	0.335	0	0.405
colera	0	0	0.588

cos(CAS,CMA) ≈ 0.94
cos(CAS,ATC) ≈ 0.79
cos(PAP,WH) ≈ 0.69

Para simplificar este ejemplo, no se realizo ponderación idf.

Referencias

- Introduction to information retrieval https://nlp.stanford.edu/IR-book/
- Jurafsky D. and Martin J. (2021) Speech and Language Processing (3rd ed. draft). Online: https://web.stanford.edu/~jurafsky/slp3/
- Yoav Goldberg (2017). Neural Network Methods in Natural Language Processing.
- In Deng, L., & In Liu, Y. (2018). Deep learning in natural language processing.