


Sistemas de Recuperación de Información

Motor de
búsqueda:
Sherlock 

Autores

Laura Victoria Riera Pérez

Leandro Rodríguez Llosa

Marcos Manuel Tirador del Riego

Resumen Se abordan los aspectos principales de una posible implementación del modelo vectorial clásico.

Palabras clave — recuperación de información (RI) · modelo vectorial

Índice general

1. Introducción	1
2. Modelación del problema	1
2.1. Documentos	1
2.2. Normalización de un término	1
2.3. Corpus	1
2.4. Modelo base	2
3. Modelo vectorial	2
3.1. Preprocesamiento	2
3.2. Recuperación de documentos	2
4. Modelo Fuzzy	2
4.1. Descripción del modelo usado	3
5. Conclusiones	3

1. Introducción

En la actualidad, con el inmenso crecimiento del internet, se convierte en un reto cada vez mayor el manejo de la información, su recuperación y la extracción de conocimiento de ella. Por esto, es de especial interés la creación de algoritmos que ayuden a su manipulación. En este informe se expondrán algunas ideas importantes seguidas a la hora de implementar un modelo de recuperación de información clásico: el modelo vectorial.

2. Modelación del problema

2.1. Documentos

Para el desarrollo de este Sistema de Recuperación de la Información modelamos un documento como un objeto que contiene al menos dos propiedades, un `doc_id` que identifica de manera única a un documento dentro del conjunto de documentos del dataset en cuestión; y un `text` que corresponde con el texto de este. También puede tener otras propiedades, por ejemplo: `title`, `author`; pero eso depende de la riqueza del dataset que provee el paquete de Python `ir_datasets` (ver [3]).

Como el texto de un documento es incómodo de manipular por venir en forma de `string`, este se tokeniza y convierte en una lista de términos indexados normalizados. Esto se logra haciendo uso del paquete de Python `re` (referirse a [4]) que proporciona una colección de funciones que facilitan el trabajo con expresiones regulares.

2.2. Normalización de un término

Es importante destacar algunas asunciones que se tuvieron en cuenta en el proceso de tokenización. No se considera relevante la diferenciación entre una letra mayúscula y una minúscula, ya que, en la mayoría de los casos, el significado semántico que expresan es el mismo. Para reducir algunos errores ortográficos, y que esto no perjudique la recuperación de un documento importante, se considera que las tildes no diferencian una palabra de otra. Se conoce que esto último no es cierto en el español, pero por el momento se está trabajando con textos en inglés.

Resumiendo, se trata de llevar todos los términos a cadenas de caracteres que contienen letras minúsculas o números.

2.3. Corpus

Un corpus no es más que el conjunto de documentos de un dataset, tokenizados y normalizados.

2.4. Modelo base

Se define un modelo base como un concepto abstracto, que engloba los comportamientos comunes que debe tener cada modelo de recuperación de información.

Cada instancia de un modelo tiene un corpus asociado a este, sobre el cual se hacen las búsquedas. Se deja en manos del programador qué preprocesamientos hacer con el corpus, en dependencia del modelo que se esté implementando.

3. Modelo vectorial

3.1. Preprocesamiento

Según la fórmula de similitud del coseno ([1, Ecuación (6.10)]), se puede notar que el aporte de un documento a la fórmula se mantiene invariante para todas las consultas, ya que este solo depende del vector que representa al documento, el cual es independiente de la consulta. Por tanto, como el corpus sobre el cuál se va a recuperar información se mantendrá estático, se calcula el peso de cada término en cada documento para así poder utilizarlo en cada consulta que se realice.

Para calcular el peso de cada término en cada documento según [1, Ecuación (2.3)], primero se necesita calcular las frecuencias normalizadas de los términos en cada documento (TF [2, Ecuación (2.1)]), y la frecuencia de ocurrencia de cada término dentro de todos los documentos del corpus (IDF [2, Ecuación(2.2)]).

3.2. Recuperación de documentos

La primera fase es similar a la del preprocesamiento de los documentos del corpus. Lo primero que se hace es tokenizar y normalizar la consulta. Luego se hallan los TFs, y se calcula el peso de cada término en la consulta. Para el cálculo de los pesos de cada término se utiliza la medida de suavizado $\alpha = 0.4$ para amortizar la contribución de la frecuencia del término (ver [2, ecuación (2.4)]). Por último se halla la cercanía entre el vector consulta y cada vector documento, utilizando la similitud del coseno entre los vectores según [1, Ecuación 6.10], y se hace un ranking teniendo este valor calculado.

4. Modelo Fuzzy

En el modelo booleano se representan las consultas y documentos como conjuntos de palabras. Al determinar que un documento es relevante a una consulta si y solo si tiene una coincidencia exacta de las palabras en la consulta solamente nos acercamos parcialmente al contenido semántico real de los contenidos. Una alternativa sería definir un conjunto difuso por cada palabra de la consulta y determinar un grado de pertenencia de cada documento a este conjunto, para luego basado en esto, poder asignar un grado de relevancia de cada documento a la

consulta dada. Esta es la idea básica detrás de distintos modelos de recuperación de la información basados en conjuntos difusos.

Se presentan a continuación algunos conceptos de la teoría de conjuntos difusos necesario para entender el modelo implementado que se describe en esta sección.

Definición 1 ([2, Section 2.6.1]) *Un conjunto difuso A de un universo de discurso U está caracterizado por una función de membresía $\mu_A : U \rightarrow [0, 1]$ que asocia a cada elemento $u \in U$ un número $\mu_A(u)$ en el intervalo $[0, 1]$.*

Las tres operaciones más usadas de conjuntos difusos se definen a continuación.

Definición 2 ([2, Section 2.6.1]) *Sea U el universo de discurso, A y B dos conjuntos difusos de U y \bar{A} el complemento de A en U . Sea además un elemento $u \in U$. Entonces,*

$$\begin{aligned}\mu_{\bar{A}}(u) &= 1 - \mu_A(u) \\ \mu_{A \cup B}(u) &= \max(\mu_A(u), \mu_B(u)) \\ \mu_{A \cap B}(u) &= \min(\mu_A(u), \mu_B(u)).\end{aligned}$$

4.1. Descripción del modelo usado

El modelo fuzzy que se seleccionó para su implementación fue propuesto por Ogawa, Morita y Kobayashi en [5].

5. Conclusiones

Este trabajo presenta una propuesta para la modelación del problema de recuperación de información. Se detallaron aspectos generales del modelo vectorial clásico, así como decisiones de diseño específicas de la propia interpretación del problema.

Referencias

1. Maning C. D.: *An Introduction To Information Retrieval* (2009).
2. Ricardo Baeza-Yates: *Modern Information Retrieval* (1999).
3. Documentación oficial: <https://ir-datasets.com/index.html>
4. Documentación oficial: <https://docs.python.org/3/library/re.html>
5. Y. Ogawa, T. Morita y K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy Sets and Systems*, 39:163-179, 1991.