Sistema de Recuperación de Información

Carlos Rafael Ortega Lezacano and Eric Martín García
Grupo C511

Universidad de la Habana

Abstract. The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be written using the *abstract* environment.

Keywords: We would like to encourage you to list your keywords within the abstract section

1 Introducción

2 Diseño del Sistema

Un sistema de recuperación de información (IRS) se compone por el cuadruplo $< D,\ Q,\ F,\ R(q_j, d_j) >$, donde D es un conjunto de representaciones de los documentos, Q es un conjunto compuesto por representaciones lógicas de los pedidos que el usuario realiza al sistema, F es un framework para modelar las representaciones y R es una función de orden donde a cada consulta q_j y un documento d_j le asigna un valor acorde a la relevancia del documento para esa consulta. El proceso de interacción de un sistema con un usuario sigue el siguiente comportamiento: Primeramente el usuario realiza una consulta (un elemento de Q), esta pasa al motor de búsqueda, este es la componente más importante del sistema ya que en este se realizan las representaciones de los documentos (corpus) que contiene el sistema a un formato el cual pueda ser fácilmente empleado por el IRS elegido, también contiene el procesador de consulta el cual realiza una transformación de la consulta, primeramente para interpretar los operadores que pudo emplear el usuario y segundo para realizar la representación interna de la consulta, por ultimo el sistema dará como resultado una lista ordenada por relevancia de los documentos del corpus. Analizaremos a continuación el procesador de consultas, como se compone el motor de búsqueda y finalmente la salida del sistema.

2.1 Procesamiento de la Consulta

La consulta es un conjunto de representaciones lógicas de las necesidades del usuario. El procesamiento de la consulta realizada por el usuario permitirá no solo transformarla a la representación interna que emplea el motor de búsqueda también permite que podamos definir consultas más complejas y por lo tanto

que satisfagan más al usuario. Podemos pensar en una consulta básica como una entidad, por ejemplo si hablamos en temas médicos *atrofia muscular* sería una entidad, si agregamos operaciones lógicas como `or` y `and` entonces contamos con más recursos a la hora de definir la consulta. Existen además otras operaciones como el uso de comillas para la expresión de forma exacta, comandos para búsqueda avanzada entre otras. En nuestro sistema hemos decidido solamente emplear las operaciones `or` y `and` debido a que son las más relevantes en cuanto a la recuperación de información de textos que es el principal objetivo del sistema, no obstante es posible agregar nuevas operaciones sin necesidad de realizar grandes modificaciones debido que para un conjunto de entidades y operaciones entre ellas, el procesador de consultas se encarga en crear un texto plano, sin operadores, el cual puede ser interpretado por el motor búsqueda.

2.2 Motor de Búsqueda

El motor de búsqueda contiene el modelo de recuperación de información, además en este se realiza el preprocesado de los documentos y su representación interna, también cada consulta se almacena en su forma interna para ser empleada en otras tareas. Además contiene la función de ranking la cual clasifica la relevancia de los documentos del corpus acorde a la consulta y el IRS.

Representación de Documentos: Para representar los documentos primeramente es necesario realizar un preprocesado, en nuestro sistema procedemos de forma simple empleando recursos del procesamiento de lenguaje natural. Las siguientes transformaciones son realizadas para cada documento del corpus:

1. **Tokenizar el documento:** Primeramente es necesario crear la representación básica de un documento, la división en tokens, estos son unidades básicas, de esta forma podemos realizar procesamiento más avanzado sobre el texto, ya desde esta parte del procesamiento podemos retirar signos de puntuación y otros símbolos que carentes de información.
2. **Eliminación de *stopwords*:** Las *stopwords* como sabemos son palabras vacias de significado, las cuales pueden servir de nexo entre entidades o funcionan como modificadores. Si incluimos estas palabras en la representación del documento se afecta la efectividad del modelo debido a la alta frecuencia que poseen estas palabras en los textos.
3. **Stemming:** Este método busca relacionar palabras con igual significado pero que difieren en cuanto a la escritura, por la aplicación de prefijos o sufijos, se encuentran en plural o singular entre otras diferencias, en este caso no usaremos lematización que es común que se emplee luego del stemming, con remover las partes correspondientes es suficiente para obtener buenos resultados.

Estas técnicas son las empleadas por el sistema para obtener un conjunto de textos formados por entidades con un significado y peso correcto para empezar a construir el modelo, pero antes de definir el modelo es necesario definir como

serán estas entidades y que representación interna tendrán. Para las entidades se decidió emplear la más básica, palabras, cada palabra constituye una entidad, sin importar de que documento del corpus sea. Una vez que tenemos el tipo de entidad pasamos a representar el corpus, el corpus se compone de todos los documentos por lo tanto emplearemos primeramente un mapping en donde cada palabra del corpus recibe un identificador, de esta forma tendremos cada elemento identificado de forma única, luego pasamos a formar un *Bag of Words*.

Bag of Words: Consiste en obtener todas las palabras diferentes que aparecen en el corpus. Obteniendo un conjunto de palabras pero perdiendo su orden secuencial. Al realizar esta operación la estructura del corpus se pierde siendo necesario guardar dicha estructura para ser empleada más adelante como parte de la salida del sistema.

El proceso de Bag of Words permite la unificar

2.3 Salida del Sistema

3 Sobre la implementación

4 Evaluación del Sistema

5 Resultados Obtenidos

6 Recomendaciones

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`