MÁSTER EN INTELIGENCIA ARTIFICIAL

INFORMÁTICA BIOMÉDICA

Buscador de LOINC basado en SVMrank optimizado usando clicks de usuarios*

Autores
AÍDA MUÑOZ MONJAS
CÉSAR PANTOJA ROSALES
GABRIEL RIVERA CÁRDENAS

December 13, 2022

1 Introducción

Tradicionalmente, algoritmos como el TF-IDF usados para realizar motores de búsqueda clasifican los resultados obtenidos para cada consulta en ”relevantes” e ”irrelevantes”. Esta clasificación no caracteriza de manera completamente correcta las opiniones de los usuarios, ya que hay resultados más relevantes que otros, de manera que se puede establecer un orden prácticamente total de la relevancia óptima de los resultados.

Obtener este ranking sin tener un feedback explícito no es trivial, y conseguir estos comentarios por parte de los usuarios es difícil. El conocimiento sobre a qué entradas de la búsqueda acceden los usuarios nos puede proporcionar información equivalente, de manera mucho menos costosa. El principal inconveniente de utilizar el conocido como ”click-through data”, datos sobre los clicks de los usuarios, es la cantidad de ruido presente en los datos y la dependencia que existe entre los clicks de los usuarios y el orden de los documentos recibidos.

Sin duda este tipo de datos son útiles y poco costosos de conseguir, pero su calidad no se puede comparar con la de juicios de relevancia generados por expertos del dominio.

En este trabajo, se nos pide implementar los algoritmos descritos en el artículo [1] sobre un set de tres búsquedas sobre la terminología LOINC.

LOINC (Logical Observation Identifiers, Names and Codes)[2] es una terminología de términos de laboratorio, donde cada concepto viene definido por el componente medido (component), el sistema sobre el que se observa (system), la propiedad observada (property) y su nombre (long common name), este último agrupando las otras tres características del término.

2 Desarrollo del buscador

Utilizando el lenguaje python, se ha adaptado el dataset proporcionado a las necesidades del proyecto, y se ha preparado una implementación de un buscador basado en el algoritmo BM25, optimizado mediante los clicks de los usuarios.

2.1 Procesado de datos y aplicación del BM25

En el dataset se proporcionaron tres búsquedas sobre LOINC, de las cuales cada consulta tenía una lista de posibles respuestas. Para cada una de las consultas se obtuvo un dataset similar al de la Figura 1.

id	loinc_num	long_common_name	component	system	property
0	1988-5	C reactive protein [Mass/volume] in Serum or Plasma	C reactive protein	Ser/Plas	MCnc
1	1959-6	Bicarbonate [Moles/volume] in Blood	Bicarbonate	Bld	SCnc
2	10331-7	Rh [Type] in Blood	Rh	Bld	Type
3	18998-5	Trimethoprim+Sulfamethoxazole [Susceptibility]	Trimethoprim+Sulfamethoxazole	Isolate	Susc
4	1975-2	Bilirubin.total [Mass/volume] in Serum or Plasma	Bilirubin	Ser/Plas	MCnc

Figure 1: Ejemplo del dataset recibido para la consulta ”glucose in blood”

El procesado de los datos de este problema se ve afectado por dos importantes decisiones de diseño: la elección del algoritmo de búsqueda, y los campos utilizados para realizar la

búsqueda. Como algoritmo de búsqueda se utilizó el algoritmo BM25Okapi, por ser uno de los algoritmos base más robustos habitualmente utilizado en este campo. Para la implementación de la búsqueda se utilizó la librería `rank_bm25` y su implementación del algoritmo. En segundo lugar, se decidió utilizar los campos *long common name*, *component* y *system* como la base de este motor de búsqueda, ya que proporcionan información suficiente para poder realizar la búsqueda, y así eliminar la necesidad del campo *component*, cuya información no se podía utilizar con las consultas propuestas.

Los autores de este trabajo seleccionaron, actuando como usuarios, a cuáles de los códigos harían click según la descripción textual de la búsqueda. A partir de esta información sobre los clicks, se puede proceder a la aplicación del algoritmo BM25 sobre cada consulta, obteniendo una estructura de datos que representa la importancia relativa de cada uno de los atributos del código de LOINC respecto a la query.

```
[ glucose in blood ] BM25 rank:
    long_common_name  component  system  index  sum_clicks
0           2.826131   2.710869     0.0      1           0
1           2.557807   2.710869     0.0      2           0
2           2.287464   2.710869     0.0      3           0
3           2.219103   1.928531     0.0      4           0
4           2.053367   2.710869     0.0      5           0
..               ...        ...     ...    ...         ...
62         -1.204574  -0.433034     0.0     63           0
63         -1.204574  -0.433034     0.0     64           0
64         -1.204574  -0.433034     0.0     65           0
65         -1.204574  -0.433034     0.0     66           0
66         -1.204574  -0.433034     0.0     67           0
```

Figure 2: Resultado de aplicar el algoritmo BM25.

Cabe destacar que con el algoritmo BM25, la importancia de los contenidos del system se marca como `0` en todos los casos. Esto se debe a que este campo contiene abreviaturas, representando por ejemplo la palabra *blood* como *Bld*, por lo que una búsqueda por palabra exacta no obtiene ningún resultado.

Como se indica en el articulo [1], la información relativa al clickthrough data se codifica como tripletas, donde cada búsqueda (query) se relaciona con los resultados obtenidos con el algoritmo de búsqueda, y el número de clicks realizados sobre cada uno de ellos.

El dataset apropiado para aplicar el algoritmo SVM rank, tiene un formato específico. En este caso, la consulta "glucose in blood" se representa como se indica en la figura 3, donde el primer campo indica el número de clicks del link, el segundo campo contiene un identificador de la consulta y los otros tres campos contienen el resultado del BM25 para cada uno de los campos estudiados.

```
1 qid:0 1:1.1981276235493101 2:-0.4330342466794794 3:0.0
3 qid:0 1:1.0121179612527342 2:-0.4330342466794794 3:0.0
1 qid:0 1:1.0121179612527342 2:-0.4330342466794794 3:0.0
3 qid:0 1:0.4879868238100544 2:1.560738709033399 3:0.0
2 qid:0 1:0.22053892859650248 2:1.9232975426567003 3:0.0
3 qid:0 1:-0.16160371801672088 2:-0.4330342466794794 3:0.0
3 qid:0 1:-0.32805449537933723 2:-0.4330342466794794 3:0.0
1 qid:0 1:-0.32805449537933723 2:-0.4330342466794794 3:0.0
1 qid:0 1:-0.47481720601819 2:-0.4330342466794794 3:0.0
2 qid:0 1:-1.2045742995485822 2:-0.4330342466794794 3:0.0
```

Figure 3: Búsqueda "glucose in blood" con el formato necesario para aplicar SVM rank.

De esta manera, se pudo generar un dataset con las tripletas habituales para este tipo de datos. Esta información se guardó en un segundo fichero, separando el dataset en datos de entrenamiento (`result/train.dat`) y datos de evaluación (`result/test.dat`).

Debido a la cantidad de datos con los que se trabaja, los dataset generados son significativamente pequeños, por lo que este proyecto se debe tomar como una prueba de concepto, y no una evaluación completa del algoritmo.

2.2 Implementación

La implementación realizada se puede observar en la carpeta adjunta `code`, que contiene los ficheros de código necesarios, así como los ficheros de entrada del algoritmo en la carpeta `code/input` y los resultados de la ejecución en la carpeta `code/result`.

La implementación realizada sigue el siguiente esquema:

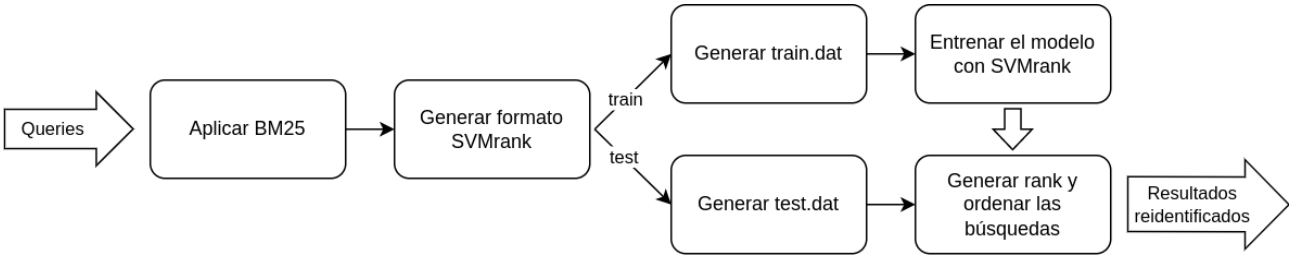

Figure 4: Esquema de la implementación

Una vez generados los ficheros `train.dat` y `test.dat` como se indica en el apartado anterior, generamos el modelo de SVMrank y realizamos la búsqueda de cada una de las consultas sobre el modelo.

Como implementación del SVMrank hemos utilizado la indicada en el artículo [1], cuyos ficheros se encuentran en la carpeta `code/aux_files`.

Este algoritmo devuelve el fichero `result/prediction_order.txt`, que indica una puntuación para cada entrada del dataset en el orden en el que se encuentran en el fichero `test.dat`, por lo que es necesario reidentificar la entrada a la que se refieren. Para esta reidentificación utilizamos el score obtenido con el BM25 para el campo *long_common_name*, ya que este campo es único para cada código de LOINC.

3 Conclusiones

En este trabajo se ha implementado un buscador de la terminología LOINC de laboratorio basado en el algoritmo SVMrank, utilizando como fuente de información los resultados de la búsqueda con el algoritmo BM25 y clicks de los posibles usuarios.

Pese a ser un ejemplo funcional de la implementación, no se pueden obtener conclusiones relevantes respecto a la calidad del algoritmo. El dataset utilizado para realizar el entrenamiento y la búsqueda no contiene cantidad suficiente de información como para conseguir métricas de este tipo.

En este trabajo se puede ver el interesante potencial de los metadatos de las búsquedas como sustituto de un ranking de experto, así como los problemas del clickthrough data.

La implementación propuesta se podría mejorar en un futuro, modificando ligeramente la entrada del algoritmo BM25, sustituyendo las siglas por su significado, o incorporando un conjunto de ”palabras equivalentes” a este set de resultados.

References

[1] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’02. New York, NY, USA: Association for Computing Machinery, 2002, p. 133–142. [Online]. Available: https://doi.org/10.1145/775047.775067

[2] LOINC. Regenstrief Institute - Home. https://loinc.org/. (Accessed on 06/12/2022).