

Relevance-Based Biomedical Information Retrieval

Final Project

Cody Valle, Jinous Esmaeili | CPEN436 | 12/09/2016

Table of Contents

**Introduction2**

**Algorithm3**

**Implementation4**

**Testing5**

**Discussion and Conclusion6**

**Contributions7**

# Introduction

This relevance-based biomedical search and ranking system that can be used to find articles relevant to a query or topic.

To calculate relevant documents, we took the user selected CUIS and compared each documents with CUIs. Ranked each document by A-TF-IDF and determined the precision @ 10 and AveP by comparing them with the Goldstandard.

For the second round we calculated the profiles for every document which is ranking the concepts by their Iw. Built the Pz profile and calculated the rank based overlap for each document.

For the last round we used the relevant documents from the RBOs calculated in previous round. We rebuild the Pz profile and ranked each document by their new RBO. Afterwards we calculated the precision @ 10 and AveP.

# Algorithm

For first two rounds we used different algorithm to implement the search.

In the first round, to get the IDF values we used HashMap to keet track of IDFs of each CUI. We Calculated TF for every CUI for every document and calculated A-TF-IDF. Stored all A-TF-IDF in an ordered list and ranked the list in a descending order.

For the second round we used HashMap to store the partial count for every sentence in Z, and calculated the necessary values to get the Iw for each document. We calculated RBOs using the maximum number of concepts from the Iw list and concepts in Z.

For the last round our Z was the relevant documents from the top 10 ranked documents from round two. We calculated RBOs using the maximum number of concepts from the Iw list and concepts in Z.

Implementation

To implement the search, we created a new class called IwStructure. This class contains a string and a (double) weight. This structure holds A-TF-IDF and document name, a concept and Iw, or document name and RBO.

We Used HashMap to keep track of IDFs of each CUI, counts of concepts, store the partial count for every sentence in Z.

We used ArrayList to track IwStructues and various types strings.

We used HashSet to develop a unique list of concepts for various purposes.

Testing

To make sure of the functionality of the search we decided to test each step as we went through.

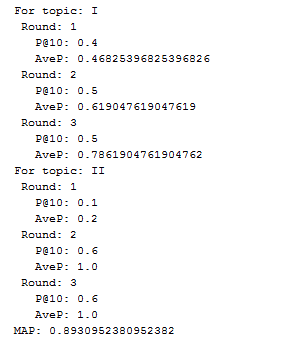
First thing we did was load up all data from files and made sure that we had them correct in memory. Then we calculated the A-TF-IDF for each document and compared our result with the previous assignment. Then we calculated precision.

Since we were sure that our A-TF-IDF and precision values are correct, we moved on to the next step. We calculated Iw for a small document that we wrote both by hand and by computer. When they both matched we moved on to the next step.

At the end we calculated the RBOs both by hand and by computer until they matched.

Discussion and Conclusion

At the beginning we did not know what a relevant document was. As a result, we couldn’t calculate precision. After asking other students and going through slides we understood what a relevant document is.

 In the context of biomedical information retrieval an issue that we encountered was the amount of time that it takes to do a search. If this program was scaled to handle a large database, the time that it takes to get results would be less beneficial.

An advantage is that we get documents relevant to our query, and a disadvantage is that there is a limit to minimizing the search time.

Contributions

Jinous and Cody started the project by going through the slides and understanding the concept behind each and every term. After understanding the purpose of project they started coding. Cody wrote the code, and Jinous made sure he was doing the algorithm correctly.

Jinous and Cody brainstormed to develop the presentation and report.