Information Retrieval Report

[Evaluation of Information Retrieval Systems]

100022010

ABSTRACT

This report outlines the assignment completed and examines the techniques used and evaluates the experiments conducted.

1. INTRODUCTION

The assignment was to produce a search engine that works over the uea.ac.uk/computing domain. Using the crawler I indexed the domain stripping out the formatting of the pages so it left me with plain text in lower case format without punctuation. Next I constructed a basic system with a simple interface to run training queries based on the TF*IDF measure. When running the queries it allowed me to collect a list of ranked results in the form of URLs.

After completing the basic system I attempted to improve my results by using a variety of different implementations. These included stemming and stop words.

In this report I will discuss in more detail the techniques I have used, and the problems and limitations of these techniques. I will then examine my results from my information retrieval systems and evaluate my experiments conducted.

2. TECHNIQUES USED

An information retrieval system processes queries entered by a user. Queries are statements of information that the user would like to enquire about. Queries don't just uniquely identify a single object in a collection, instead they can identify several objects that could match the query with varying degrees of relevancy.

Most information retrieval systems compute a numeric score showing the top ranking objects. These are presented accordingly to the user in a ranked list depending on how well each object returned matched the query. I do this by calculating the normalised TF*IDF score and the cosine similarity.

After completing the basic system I attempted to improve my results by using Stemming. This process reduced the words down to their basic stem or root form. Stemming is used to improve retrieval effectiveness and to reduce the size of indexing files. Another improvement to the system I made was the introduction of Stop Words. Stop Words are a list of pre-defined words that are filtered out before or after processing text. They are usually the most common words used in a language. For example "the, of, a, is, at, which". I used a set list of 30 of the most common words used for querying.

Information Retrieval evaluation focuses on measuring the performance of the system's ability at retrieving relevant information, and not the query's ability. However, the effectiveness of a retrieval system is strongly influenced by the quality of the query submitted. When devising my set of test queries I looked to build on the set of training queries and highlight the improvements that I made through the use of stemming and stop words.

2.1 Problems and Limitations

When indexing the UEA Computing domain, occasionally the domain's server would restrict access depending on how many requests to access the site were occurring. This would result in the indexer being blocked from accessing specific URLs.

A series of invalid URLs when indexing would inhibit results, as the amount of documents returned for queries would occasionally not return all the relevant results expected. Although this would still allow evaluation of results to happen, it is always good to have a full corpus of documents to evaluate from for the best results.

2.2 TF-IDF Formula

When researching how to calculate the TF*IDF formula I found two different methods for calculating the normalised TF within the support material provided. One formula used

$$TF = 1 + \log(termFrequency) \tag{1}$$

and another used,

$$TF = \frac{termFrequency}{corpusTotal} \tag{2}$$

Term frequency meaning the frequency of a specific term in a document, and corpus total meaning total number of documents within the corpus. For sake of simplification I shall simplify the formulas to $1 + \log(tf)$ and $freq/P$

This provided a lot of confusion in which was the correct formula to use, so I decided to implement both individually in their own series of systems. After implementing each different TF*IDF I could then calculate the cosine similarity to retrieve list a of ranked results. I calculate the similarity metric between each entry in a query and the associated document vector. The metric co-relates the weighted words $\mathbf{A}$ with a document vector $\mathbf{B}$ representing the normalised term frequencies.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{3}$$

A high cosine value indicates that an entry is closely related to the query and thus a good candidate for being relevant.

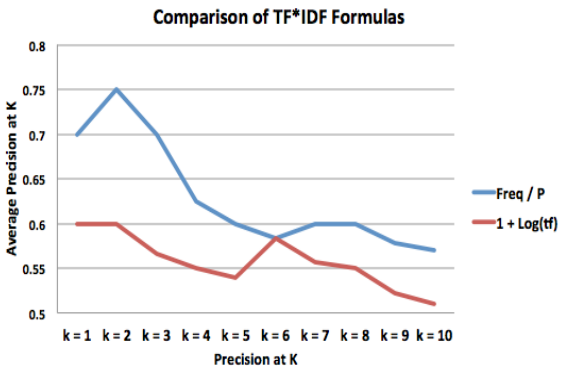

Figure 1: A comparison of TF*IDF formulas on a basic system.

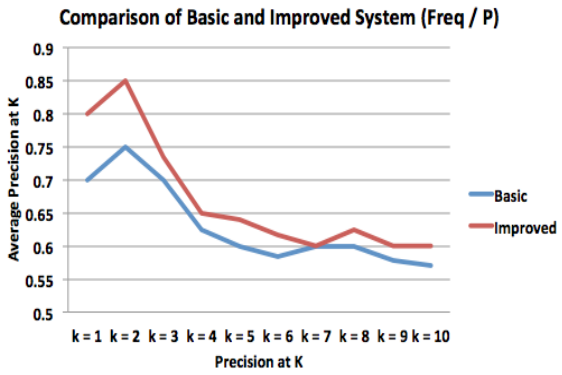

Figure 2: A comparison of the first basic system against the improved stemming and stop words system. Both using the *freq / P* formula.

3. IMPROVEMENT EXPERIMENTS

With the different TF formulas in mind, I began testing my basic system. When indexing, I stripped all punctuation and html code, then converted all text to lower case format. This is all to aid with indexing the site and with getting just the words and their accurate term frequencies. Once indexing had finished, I could then query each system with a list of suitable queries provided. From the results generated I could calculate the precision at k,

$$P(k) = \frac{RelevantDocumentsRetrieved}{k} \tag{4}$$

and the average precision at k.

$$avg.P(k) = \frac{\sum_{i=1}^{queries} P(k)_i}{queries} \tag{5}$$

This experiment showed that both formulas had the same average precision at 6. This was the closest the $1 + Log(tf)$ formula got to surpassing the $freq/P$ formula, however in comparison of the two TF formulas, the $freq/P$ formula overall returned the most relevant results in every other average precision at k. (See Figure 1)

When judging relevancy, I defined what a relevant URL was based on the pooled results provided by Dr. Dan Smith, however if the query was not ranked in relevance from his results then I'd judge to see whether there was more than 50% of the query in the document. Relevance is subjective to the user who queried the system. Something that could be relevant to me might not have been considered relevant to someone else. Taking this into account, the relevancy was decided in a consistent manner across all experiments I conducted, so that the best possible accuracy was the result.

After selecting the best TF formula, I wanted to improve my results so I could improve retrieval effectiveness and to reduce the size of indexing files, as indexing took rather a long time to complete and sometimes could present invalid URLs if the domain got overwhelmed with requests. One method of doing this is by using stemming. Stemming works by stripping the word down to its root form. (See Table 1). I implemented stemming using the UEA Lite stemmer. Although the results retrieved showed significant improvement to the system, I believe that with a system that was querying a less technical database of terms, the results could have shown a higher relevance of results.

Table 1: Example Queries for Testing Improvements

Query	Returned
Software Engineering	software, engineer
What is Java Programming	java, program
Algorithms for Bioinformatics	algorithm, bioinformatic

Another improvement to the system - on top of stemming - that I implemented was the addition of stop words. Stop Words are a list of pre-defined terms that are filtered out before or after processing the text. They are usually some most common words used in a language. The list that I used contained 30 of the most common words used in search engine querying.

I made the assumption that with the addition of stop words, my system would show noticeable differences in ranked results. However, this was not the case. When I compared the cosine similarity of the results for the system with stop words with the system with stemming you could not see any noticeable difference in the ranking of URLs. But, a very minute change in score that averaged a difference of 0.0000006 between a system with just stemming, and a system with stemming and stop words.

Following these improvements I decided to compare results from the basic system to my improved system with stemming and stop words. From the experiment I could see that the improvements made had increased the relevance of returned URLs. (See Figure 2)

4. RESULTS

Whilst conducting my improvement experiments I could see that the TF formula I chose and the changes I made had a positive effect on the system developed, and the results shown reflect that. The overall ranked URLs retrieved from querying had an average precision at 10 of 0.6, which was higher than the previous systems I developed. (See Figure 3)

I could then take these results and compare them with other systems to see how well my system weighted itself. The other systems I chose to compare with were Google and the pooled average results collated by Dr. Dan Smith from students in the class. (See Figure 4)

The results showed that my system was slightly below

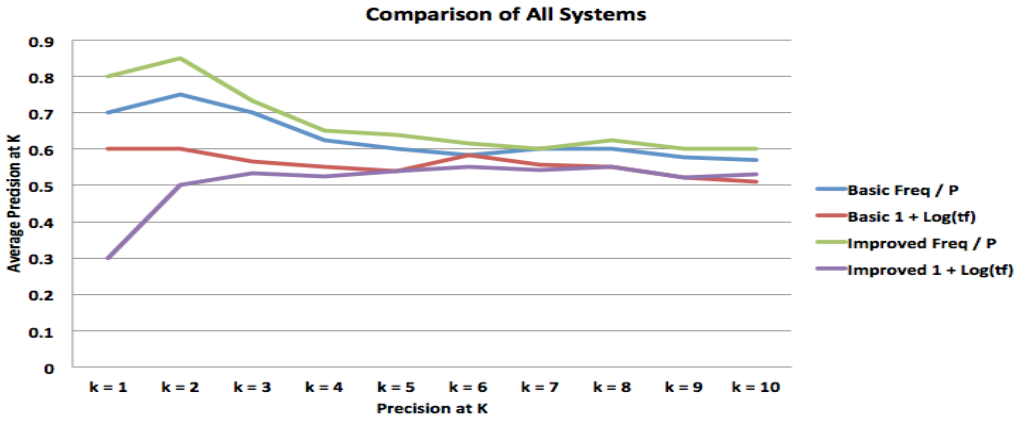

Figure 3: A comparison of all the systems I developed.

the average precision at 10 of 0.66 for both Google and the pooled results. (See Table 2). With additional improvement experiments for instance, title weighting and the use of N-grams, I believe that the average precision at 10 for my system would be higher. This is because specifically, title weighting adds extra relevance weighting to headers, meaning queries like "postgraduate course" would return far more relevant results, compared with results that contained the terms "postgraduate" and "course" separately.

4.1 Problems and Limitations

When conducting experiments, I did encounter some limitations with the results I received. The fact that I could not do recall was a problem because recall gives a more in depth feedback to how well your system works. As we only had to look at the first 10 URLs retrieved this was not possible to experiment.

Also the system does not take into account the scales of relevancy. The way I conducted my relevancy was based on the pooled results provided by Dr. Dan Smith, however if the query was not ranked in relevance from his results then I'd judge to see whether there was more than 50% of the query in the document. This however is a big limitation of relevancy because, for example the query "Tony Bagnall" there is no context attached to the query so anything matching this query will be selected as relevant. Additionally, relevance is subjective to the user who queried the system. Something that could be relevant to me in the query "Tony Bagnall" might not be considered relevant to someone else, as I know what results I want, the context would not have been specifically declared for other users to judge the relevance accurately. When designing the system again, I would have to take these limitations into consideration.

5. CONCLUSION

In this report, I have performed an empirical study examining the effectiveness of my information retrieval systems, in order to characterise the distribution of retrieval effectiveness.

If I were to do this project again, I would implement title weighting as this would be a method to increase the amount of relevant results returned to the user. Title weighting favours results that have the query terms in the title. I could also change the way I index the website, choosing to strip

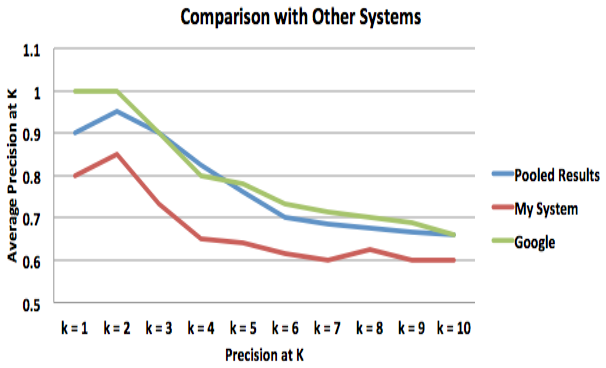

Figure 4: A comparison of my system against other systems including Google and the Pooled Results.

Table 2: Average Precision at 10 for each System

System	Average Precision at 10
Basic 1+Log(tf) System	0.51
Basic freq/P System	0.57
Improved 1+Log(tf) System	0.53
Improved freq/P System	0.6
Google	0.66
Pooled Results	0.66

the content within the body division tags of the HTML, instead of the entire site. Additionally, I could also implement n-gram methodology for predicting in a sequence. The benefits of n-gram models are the simplicity and the scalability enabling small experiments to scale up efficiently to suit the query's requirements. Also n-grams incorporate knowledge of the context of a word, which is extremely useful when judging relevance.

In conclusion, I found that the relevance of results heavily depended on the contextual information behind the query as judging relevancy is subjective. Despite this, the information retrieval systems I developed handled the queries well and the improvements made proved to produce suitable and relevant results.