MSC COMPUTER SCIENCE CONVERSION

Premlinary Research Report

COMP 30440 SOFTWARE ENGINEERING PROJECT

Student Number: 12255080 Name: David Foy

6/21/2013

Aggregation Technique

The aggregation technique which has been chosen for the meta search engine is the Borda Fuse model.

Description of technique

The Borda-Fuse model is based on a political election strategy name Borda Count.

In this model the highest ranked document in each result list returned by a search engine is given a specific number of "Borda" points (Madani, Dourado, Correia, Filipe, 2012, p. 198).

How it works...

Suppose there are n search engines, given a query, each of them will return a set of relevant pages, and suppose altogether there are c pages in the result set. For each search engine, the top ranked page is given c points, the second is given c-1 points and so on. If one page is not ranked by a search engine, the remaining points are divided evenly among all the unranked paged. Then we could calculate the total points earned by each page, and rank all the c pages in the descending order. It is a very simple procedure but it has been proven to be efficient and effective (Li, Feng, Pei, Wang, Zhou & Zhu, 2009, p.188).

This model does not require training data or the RSVs and its algorithm is simple and effective (Diaz, E. D. De, A. Raghaven, V.V, 2004).

Motivation

The main motivation behind choosing the Borda Fuse method is that is it supposedly one of the easier models to implement. It does not need to know the content relevance of web pages. The ranking output by the search engine and its overall quality seem to be sufficient for merging the results (Levene, 2011, p. 170).

Implementation Outline

The Borda Fusion method will be held in a separate function which will be called upon when one or more search engine is selected.

Bozkurt, Gurkok and Ayaz (No Date), have given an example of how the Borda Fusion method can be implemented.

The Borda count (BC) of a document *i* is computed by summing the Borda count values in individual systems(*BCA* in system *A*, etc.) as follows:

BC(i) = BCA(i) + BCB(i) + BCC(i) + BCD(i)

Now we compute the BC for each document:

BC(a) = 4 + 7 + 6 + 7 = 24

BC(b) = 7 + 6 + 3 + 2 = 18

BC(c) = 5 + 5 + 7 + 2 = 19

BC(d) = 6 + 1.5 + 2 + 6 = 15.5

BC(e) = 2 + 1.5 + 4 + 2 = 9.5

BC(f) = 2 + 4 + 5 + 4 = 15

BC(g) = 2 + 3 + 1 + 5 = 11

Sorting the scores we have found in non-increasing order,

the final ranked list of documents is:

a > c > b > d > f > g > e

Review of Query Preprocessing in Search Engines

1. **Tokenizing:** Organising the results and normalising the data in a certain way to match the users query.

2. **Parsing:** Since users may employ special operators in their query, including Boolean, adjacency, or proximity operators, the system needs to parse the query first into query terms and operators (liddy, 2001).

3. **Stop Word Removal:** This step helps save system resources by eliminating from further processing, as well as potential matching, those terms that have little value in finding useful documents in response to a customer's query. The stop list might also contain words from commonly occurring querying phrases, such as, "I'd like information about." However, since most publicly available search engines encourage very short queries, as evidenced in the size of query window provided, the engines may drop these two steps (liddy, 2001)..

4. **Stemming:** Removes word suffixes, perhaps recursively in layer after layer of processing. The process has two goals. In terms of efficiency, stemming reduces the number of unique words in the index, which in turn reduces the storage space required for the index and speeds up the search process. In terms of effectiveness, stemming improves recall by reducing all forms of the word to a base or stemmed form. For example, if a user asks for *analyze*, they may also want documents which contain *analysis*, *analyzing*, *analyzer*, *analyzes*, and *analyzed*. Therefore, the document processor stems document terms to *analy-* so that documents which include various forms of *analy-* will have equal likelihood of being retrieved; this would not occur if the engine only indexed variant forms separately and required the user to enter all. Of course, stemming does have a downside. It may negatively affect precision in that all forms of a stem will

match, when, in fact, a successful query for the user would have come from matching only the word form actually used in the query (liddy, 2001).

5. **Synonym Matching and Query Expansion:** Since users of search engines usually include only a single statement of their information needs in a query, it becomes highly probable that the information they need may be expressed using synonyms, rather than the exact query terms, in the documents which the search engine searches against. Therefore, more sophisticated systems may expand the query into all possible synonymous terms and perhaps even broader and narrower terms (liddy, 2001).

References

Diaz, E.D. & De, A.& Raghaven, V.V. (2004). On Selective Result Merging in a Metasearch Environment. Uregina. Retrieved June 21st, 2013, from http://www2.cs.uregina.ca/~wss/wss04/04/wss04-52.pdf

Madani, K. & Dourado, A. & Correia, R. A & Filipe, J. (2012). Computational Intelligence. USA: Springer.

Li, Q. & Feng, L. & Pei, J. & Wang, S. X. & Zhou, X. & Zhu, Q. M. (2009). Advances in Data and Web Management. Berlin: Springer

Levenne, M. (2011). An Introduction to Search Engines and Web Navigation, 2nd Edition. Hoboken, New Jersey: John Wiley & Sons, Inc.

Liddy, E. (2001). How a Search engine Works. Info Today. Retrieved June 21st, 2013, from http://www.infotoday.com/searcher/may01/liddy.htm

Bozkurt, I. N. & Gurkok, H. & Ayez, E. S. (No Date). Data fusion and Bias Performance evaluation of various data fusion methods. (Unpublished doctoral dissertation). Bilkent University. Ankara, Turkey.