Automatic Relevance Detection & Feedback in a Web Search Engine

Mark Grenham CASE4 53379485 m@rkgrenham.com

Disclaimer

A report submitted to Dublin City University, School of Computing for module *CA437: Multimedia Information Retrieval, 2007/2008*. I hereby certify that the work presented and the material contained herein is my own except where explicitly stated references to other material are made.

Abstract

This document outlines the functional specification for an Implicit Relevance Feedback (RF) implementation in an existing Information Retrieval (IR) System focusing on web search IR systems. The proposed relevance feedback implementation passively monitors users activities with search results and automatically updates the original query results with new ones based upon the output from the relevance feedback algorithm. This document also outlines the architecture of such a system and also the best algorithms and techniques chosen and the reasoning behind these choices.

Overview

Implicit feedback techniques in an Information Retrieval (IR) system are the more attractive method for Relevance Feedback (RF) as they do not rely on any additional input or effort from users. This is particularly true in the case of web search engines which by their very nature have the largest set of indexed documents and users. The proposed RF implementation in this document monitors the users behavior and uses this information to update the original web search query and also the results page thus implementing Implicit Relevance Feedback.

Research [1] shows that the time a user spends reading a document is a major factor in determining the users preference for that document and in the context of an Information Retrieval system can be used to determine the relevance of that document to the initial query. This is the foundation of the Relevance Feedback for the Information Retrieval system outlined here.

Another factor that had to be considered was which Relevance Feedback algorithm is best suited to this type of web search IR system. The three main RF algorithms examined are *Rocchio [2]*, *Robertson/Sparck-Jones(RSJ) [3]*, and *Bayesian [4]*.

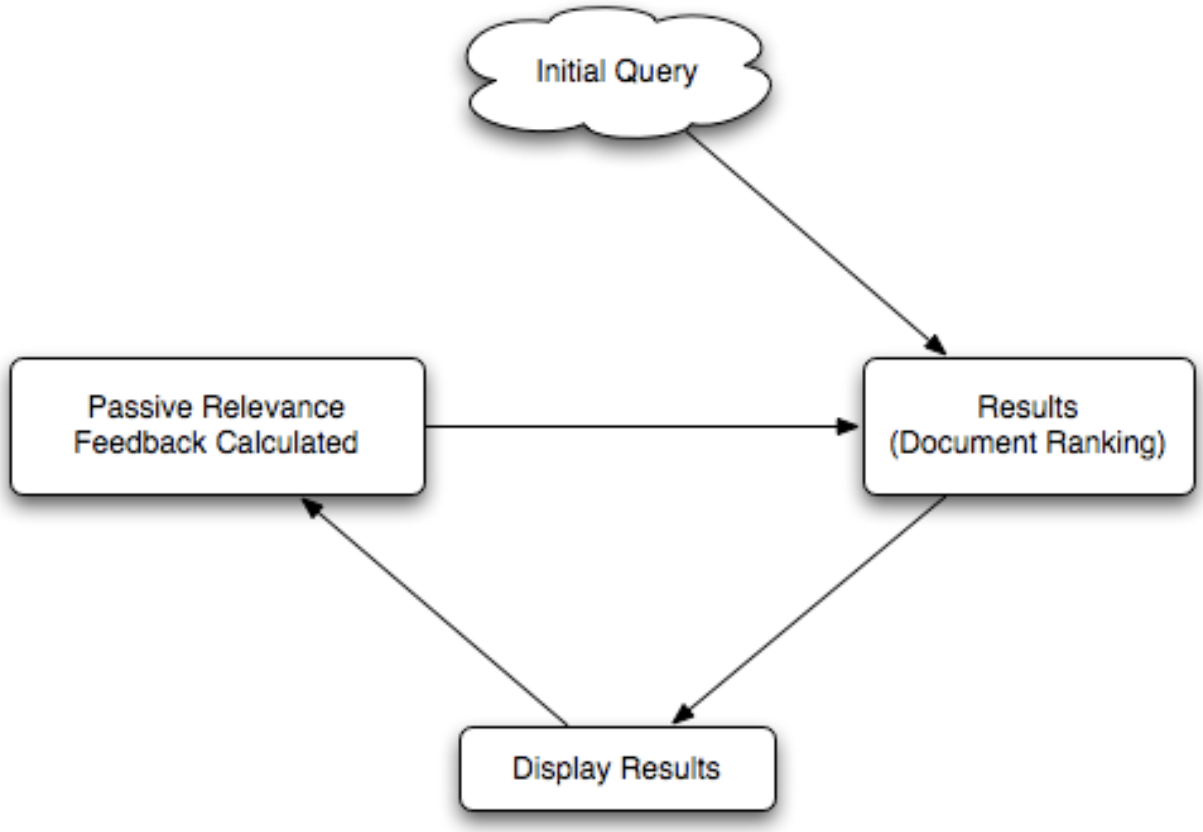
Since the web search engine frontend is web based the data that we can collect is very limited without having additional software installed on the users computer. This is a limitation that will have to be overcome without requiring users to install additional software.

Functional Description

As mentioned above Implicit Feedback has been chosen as the Relevance Feedback technique as it does not require any additional effort on the users part which is ideal for web search. The user behavior we will be observing will be time spent reading a document.

$$\underline{\textit{Time Spent Reading}} = (\ \underline{\textit{Time Closed}} - \underline{\textit{Time Opened}}\ )\ \textit{Seconds}$$

This is about as accurate as we can get to calculating the time spent reading without installing additional software on the users computer. However this is sufficient for this particular project.



The Bayesian [4] RF algorithm was selected as research [5] has shown that it performs best with web search compared to RSJ [3] and Rocchio [2]. Also unlike Rocchio or RSJ Bayesian is a relevance feedback algorithm and so can be used independently of the retrieval method used which is ideal for this project as we are only concerned with relevance feedback with an existing IR system (i.e. a web search engine). Bayesian also does not require a query, it only requires user feedback in the form of *relevant* or *not relevant.* With this in mind we can calculate the relevant documents with the following:-

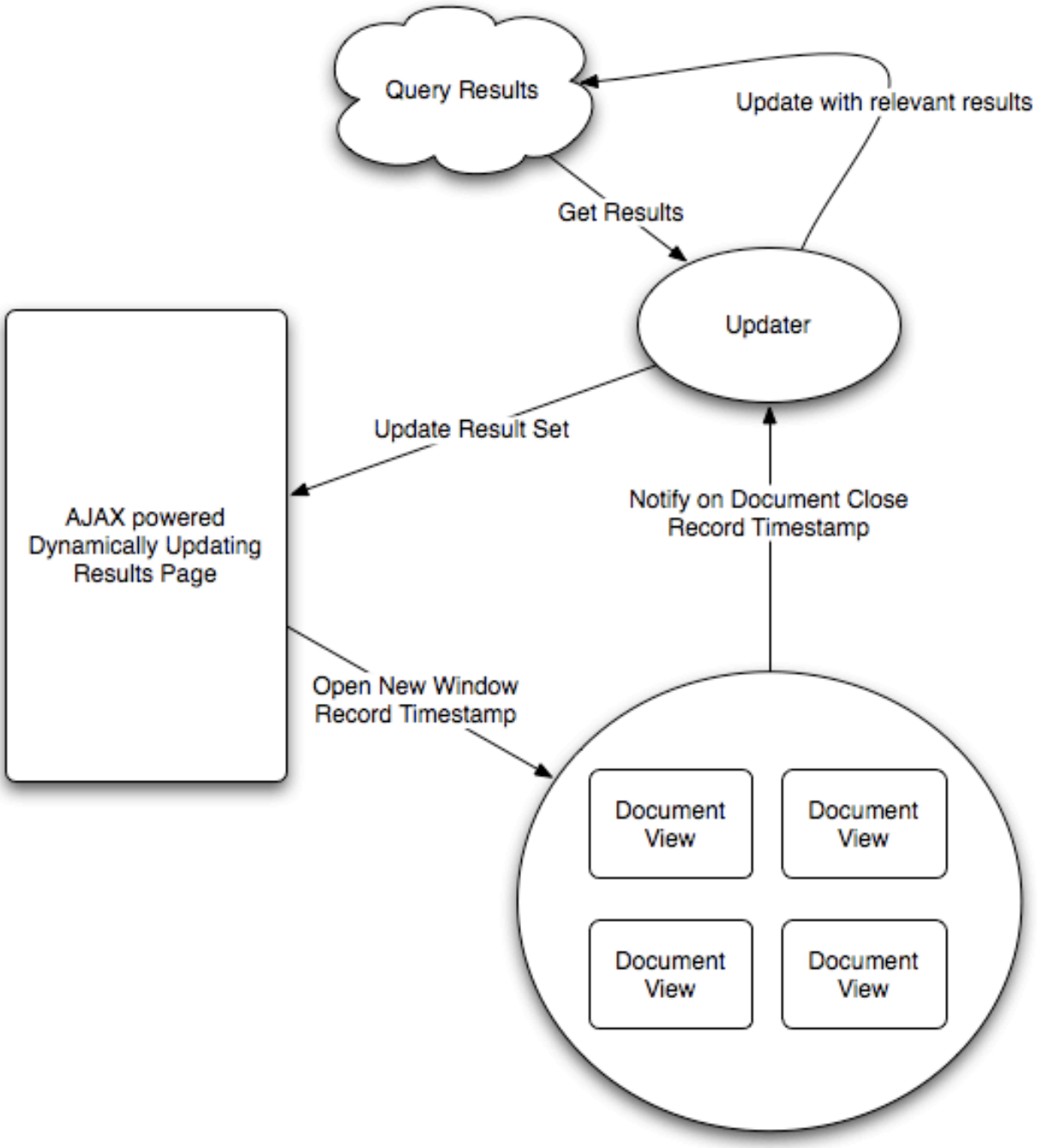
$$\underline{T} = (\ \text{average}(\ \underline{\text{Time Spend on all opened documents}}\ )\ /\ 2\ )$$

$$\underline{\text{Relevant Documents}} = (\ \underline{\text{unopened documents}}\ ) + (\ \underline{\text{opened documents}}\ \text{where}\ \underline{\text{Time Spent}} >= \underline{T}\ )$$

all others are marked not relevant.

This data is then plugged into the bayesian algorithm for the current iteration. The results page is then updated automatically with the new results set.

In order to calculate the time spend reading a document the time opened is recorded when the user clicks on the document link from within the results page. Capturing the time closed is a more difficult task however this can be captured by opening the requested document in a new window encapsulated in a single HTML frame. Embedded in the frame container would be some javascript that will trigger an event recording the time the document was closed. Once time closed has been received, calculations are made and plugged into the Bayesian relevance feedback algorithm. The search page is then automatically updated with the new results seamlessly via AJAX.



The data used in relevance feedback within this system is limited observing only one factor (time spent reading an article) however the benefits of not having to install additional software on the users computer far outweigh the cons of having only one major factor for feedback. The relevance feedback of this system will also rely on the user having a Javascript capable web browser and that Javascript is also enabled. If a browser does not support Javascript or Javascript is disabled the user will not be able to utilize the relevance feedback system outlined here.

The assumption that the existing web search engine is already implemented and working efficiently and as such the details of the hardware and retrieval related components are out of the scope of this document.

Implementation & Evaluation

The relevance feedback system outlined in this document should not be difficult to implement as some of the more complex aspects of an Information Retrieval system will be in the retrieval components itself. The RF system could be implemented in the same language as the IR system. The only new factors that the RF will implement is an AJAX interface and a code implementation of the Bayesian RF algorithm.

For testing purposes the Relevance Feedback system could be implemented in a rapid development web adapted language such a Perl, PHP, Ruby or Python. The IR system could be implemented by utilizing an existing search engines search *Application Programmer Interface* (API) such as the Google Search API. A group of test subjects could be given a list of topics to research and some would gave to use the existing IR system and the other half could use the new RF system sitting on top of the IR system. Each user should be timed and marked on a predefined list of requirements. A comparison can then be completed on the resulting data. Comparison methods similar to those used to determine the optimal RF algorithm [5] for plotting and gauging the result set from the IR system on its own and also from the IR system with the new RF component sitting on top of it.

A survey afterwards of the test subjects could be used to gauge the users perception of relevance as well as plotting the results of each individual system (IR and (IR + RF)).

References

[1] - Morita, M., & Shinoda, Y. (1996) *Information filtering based on users behavior analysis and best match text retrieval.* Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 272-281.

[2] - Rocchio, J. (1972) *Relevance feedback information retrieval.* In Gerard Salton (ed.): The Smart Retrieval System - Experiment in automatic Document Processing, pp. 313-323. Prentice-Hall, Englewood Cliffs, N.J.

[3] - Robertson, S.E., Sparck-Jones, K. (1976) *Relevance weighting of search terms*. Journal of the American Society for Information Science 27, pp. 129-146.

[4] - Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V., and Yianilos, P.N. (2000) *The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments.* IEEE Transactions on Image PRocessing, 9(1): 20-37

[5] - Vinay, V., Wood, K., Milic-Frayling, N., & Cox, I.J. (2005) *Comparing relevance feedback algorithms for web search.* http://www2005.org/cdrom/docs/p1052.pdf