A Model of Relevance Feedback for Distributed Information Retrieval

V.V. KLUEV
The Core and Information Technology Center
University of Aizu
Tsuruga Ikki-machi Aizu-Wakamatsu City
Fukushima 965-8580
JAPAN
vkluev@u-aizu.ac.jp

Abstract: The aim of the relevance feedback model presented here is to apply accumulated users' knowledge in searching for text information. The information retrieval system keeps individual feedback from users, determines appropriate documents and expands the initial user queries using terms from titles of these documents. Preliminary tests showed positive results.

Key-Words: Relevance Feedback, Search Engine, Distributed System

1 Introduction

An exponential growing amount of electronic information is becoming widely available on the Internet and in digital libraries. A goal of information retrieval is to provide effective and efficient methods of representing, managing, retrieving and displaying such information. There are a plenty of explanations on the net concerning how to search. Two of them can be found at [1, 3]. Their aim is to teach users. However, modern information retrieval (IR) systems have to extract, accumulate and use knowledge and experience from the interaction process with users. In other words, the systems have to learn from users also.

Relevance feedback is the most popular technique in information retrieval to extract user knowledge. A common scenario to use an IR system is as follows:

1. The user submits an initial query to the IR system.
2. In response to the query the system presents to the user list items characterized by retrieved documents. Each item includes a set of components:
   - URL;
   - The title of the document;
   - A short description of the document (the aim of this item is to help the user to estimate the document);
   - A document score calculated according the measure applied on the server;
   - Auxiliary information: specification of the document format, date of indexing, document size, etc.
3. The user can mark items representing documents relevant to the topic of interest from his point of view.
4. On the basis of this information the system refines the initial user query utilizing one of the methods (examples can be found in [7, 8])
5. Steps 2 – 4 can be repeated several times.

Usually this process is stateless: the system does not take into account user's responses from the previous iterations; it does not pay any attention to responses of other users either.

In the case of a distributed search or a search inside narrow topic specific collections, the aforementioned scenario does not work well. The system should collect somehow knowledge from the users' response and apply it to improve the search. Authors of study [2] noted, that although users can make explicit relevance judgments concerning why a document may be relevant to an information need, current systems have little means of using this information.

In this research, we propose a new model of relevance feedback for distributed information retrieval systems. Its aim is to take into account collective relevance judgments of users and apply them to the search process.

In the next part of this paper (Section 2), we explain the main components of the aforementioned model. Then we describe its key mechanisms (Sections 3). After that, we present results of our preliminary experiments (Section 4). Final remarks conclude the article.

2 Model: Main Ideas

The idea behind our model is very simple: We propose to retain the following information from each document marked by a user as relevant:

- URL;
- The title of the document (stop words should be eliminated);

The initial query should be retained as well. The system should count the frequency of submitted topic related queries and have a counter for each marked relevant document. Terms from the title of the often-marked relevant documents should be used to expand the current user query. These terms may be regarded in relation to the query. They can be useful in retrieving relevant documents. Several documents from the set related to the query should be presented to the user as result of the search. This is accumulated knowledge of the system about the current user query.

The following tasks should be resolved to implement the aforementioned ideas:

- Determine topic related queries.
- Select a tool to keep, update and access the necessary information.
- Choose decision-making mechanisms to select appropriate terms to expand queries.

Our solution of the aforementioned tasks has been presented in the next section.

3 Key Mechanisms

We applied a very simple approach to determine topic related queries. It was shown [6] that the average length of a user query is equal to 2.7 words. We retained only queries consisting of one, two or three words. Simple string matching was used to make a final decision about a query topic. A new two-word query inherits all components from any one-word predecessor if this word is a substring of a given query. A new three-word query usually inherits components from two-word predecessors.

LDAP (Lightweight Directory Access Protocol) service [5] was selected as a temporary solution to keep all information about queries. This directory service is a specialized database optimized for reading, browsing and searching. Directory updates are typically simple. LDAP service has been used as a tool of source selection in a distributed search system like OASIS [4]. Each OASIS server keeps its local LDAP database, and the global database exists as well. The following algorithm is used to process the user query (we assume that the length of the query does not exceed three words):

1) Receive the user query.
2) Send the request to the local LDAP server with the first word from the query.
3) Repeat step 2 with the next word, if the response is zero (it means that this word is new in this database).
4) Using simple string matching, determine the most appropriate reference, if the respond is not zero (the LDAP server can send up to three references; one of them is a reference to a query consisting of one word; the second one is related to a two word query and a three word query corresponds to the last reference).
5) Receive an URL of an often-marked relevant document and its title from the LDAP server.
6) Expand the initial query with terms from this title.
7) Process the query (in a general way).
8) Merge the results of the search with the URL obtained at step 5.
9) Present results to the end user.
10) Receive the feedback from the user (user marks the relevant document from the list presented to him).
11) Send a request to the local LDAP server with the initial query and with the components of the marked documents to include them into database or to alter counters connecting with these documents if they were already put into database.
12) Repeat steps 6 – 11 (Step 8 should be skipped).

We use the cookies mechanism to save the initial user query.

If the local LDAP server does not contain any information related to the query terms, then requests are sent to the global server.

Local databases are periodically merged with the global base.

4 Testsbeds

The proposed approach has been tested using the OASIS system. These tests are discussed in this section. The following configuration of the system was dedicated for our experiments: Three local servers (the installation place being Aizu Wakamatsu City) and one global LDAP server (at Koriyama City). The distance between these cities is about 60 km. The test topic specific collections [4] consisting of the real Internet data were involved in our experiments. Table 1 describes a distribution of document collections installed on the servers.

Table 1 Location of the collections

Servers	Collections	Number of documents
Aizu: 1	Programming Languages	7659
Aizu: 2	Algorithms	7775
Aizu: 3	Travelogues Linux & Unix Information Retrieval Research Groups Physics Card Games Museums Monitors	226 488 202 811 467 798 444 70

We used a set of short queries, similar to those submitted to the search engine. They consist of one, two and three words. These queries reflect a real search process on the Web. We selected them from the log file generated during the On-line Aizu Internet – Search contest held at the University of Aizu in 2001. The task for each participant in this contest was to find answers to questions. Participants had to formulate the queries and to submit them to the search system. Table 2 presents several examples of the questions and queries obtained from them.

Table 2. Query Examples

Questions	Queries
Malta is a small country in the sea. Could you write its name?	Malta
Nowadays we cannot imagine a computer without a mouse. Who has designed a mouse?	Mouse designer
When was the first artificial satellite of the Earth launched?	First artificial satellite

The track of the search made by participants was reconstructed using a log file. We simulate a search using the presently being discussed algorithms. To meet information needs, participants made three to four iterations on average. In the case of using our model, first, we taught the system submitting several topic related queries; after that we found right answers more quickly: The average number of iterations was equal to one and two.

5 Conclusion

This paper introduces a model of relevance feedback, which can be applied to distributed search systems. The idea behind this model is to accumulate and utilize knowledge from the users about information needs. Users usually submit poor queries: they are very short. The system tries to guess the topic of interest expanding the query using accumulated relevance feedback from the previous users. Preliminary tests showed promising results: After the teaching period the system can retrieve results more accurately.

More experiments are needed to carefully observe the effect of automatic query expansion used in our model.

References:

[1] http://websearch.about.com/cs/howtosearch/

[2] N. Denos, C. Berrut, and M. Mechkour. An Image System Based on the Visualization of System Relevance via Documents. In Proceedings of the 8th International Conference on Database and Expert Systems Applications, France, 1997.

[3] http://searchwebmanagement.techtarget.com/sDefinition/0,,sid27_gci212955,00.html

[4] V.Kluev, Source Selection in a Distributed Search System, in V.Kluev, N.Mastorakis, editors, Topics in Applied and Theoretical Mathematics and Computer Science, WSEAS Press, 2001, pp. 293 – 298.

[5] http://www.openldap.org/

[6] http://www.yandex.ru/yandex_history_engl.html

[7] David A. Grossman and Ophir Frieder, Information Retrieval, Kluwer Academic Publishers, 2000, 254 p.

[8] Djoerd Hiemstra, Stephen Robertson, Relevance Feedback for Best Match Term Weighting Algorithms in Information Retrieval. In Proceedings of the Second DELOS Network of Excellence Workshop on "Personalization and Recommender Systems in Digital Libraries", Dublin City University, Ireland, 18-20 June 2001.