GES 2007 German e-Science

Available online at http://www.ges2007.de

This document is under the terms of the
CC-BY-NC-ND Creative Commons Attribution

SOME RIGHTS RESERVED

A Multi-Agent Framework for Personalized Information Filtering

A. Lommatzsch, M. Mehlitz and J. Kunegis

DAI-Labor, TU Berlin, Franklinstraße 28/29, 10587, Berlin, Germany
email: {andreas, martin.mehlitz, kunegis}@dai-labor.de
phone: (+49 30) 314 25318, *fax:* (+49 30) 314-21799

Abstract

As today the amount of accessible information is overwhelming, the intelligent and personalized filtering of available information is a great challenge. The main problems are that the relevant information is spread over a big number of sources and useful information is hidden under the huge amount of useless data. To cope with this problem several filtering and information query strategies have been developed but they are usually specialized on a bounded problem and do not take into account the individual preferences of the user. Moreover most search engines rate every document separately and do consider the relationship between the documents in the result set. In this paper we present a multi-agent system that integrates heterogeneous information sources, a big number of filtering and rating strategies as well as strategies for combining ratings from different agents and optimizing the filter result set according to the individual user preferences. In the framework each information source, filtering strategy and optimization strategy is presented as an intelligent agent so that the system is open and extendable at runtime. The framework monitors the resource demand of each agent as well as the availability of system resources for choosing the most adequate agents according to the requested response time. User feedback is collected and used for optimizing the filtering strategies and for learning in which context which strategy performs best. The filtering framework provides the basis for the Personalized Information System. The first evaluation results show that the filtering framework provides better results and that new filtering strategies can be seamlessly integrated.

1 Introduction

Nowadays, desired information often remains unfound, because it is hidden in a huge amount of unnecessary and irrelevant data. For finding all the relevant data it is usually necessary to analyze potentially relevant documents from a big number of sources, rate each document based on several different strategies and deliver the user only the most relevant result, taking into account the individual user preferences, the query context as well as knowledge about sources and ratings provided by other users.

An intelligent meta-search system should integrate a big number of sources including web search engines, domain specific portals, catalogues/directories and databases. Due to the fact that the set of relevant sources is often changing the system must allow the fast and simple integration of new sources.

Another problem of meta-search engines is that usually each source uses its own search system rating strategy so that the results from several sources cannot be combined easily because of the different rating methods (and meanings) applied by each source. Moreover most filtering systems are limited to a content-based search but do not consider user ratings or individual user preferences. Promising methods of social software like tagging, rating or the evaluation of implicit feedback are often missing. An intelligent combination of alternative rating methods provides the basis for filtering out irrelevant or malicious documents (like spam or sponsored content).

Beside rating documents separately a filter framework should be able to consider the relationship between the documents in the result set, ensuring a predefined diversity, considering different authors, sources, domains and (maybe) languages. Additionally, the system should learn from user feedback and adapt to the individual user preferences.

2 Motivation

Traditional search engines usually only consider one type of source (web pages) and do not consider the search context. Using the example of some typical scenarios we derive the requirements for next generation personal information assistant.

- Depending on the user's information demand different information is potentially relevant. This means that an intelligent information system must be able to select the most relevant information source (transparently to the user). E.g. a basic introduction to a topic of interest can often be found in reference book or in tutorial provided by schools and universities. Scientific research paper can usually be found in databases hosted by specific conferences.
- Existing search engines usually use only limited data collections. Due to this fact, users often have to spend much time on trying several search engines and on learning the special query syntax. Next generation information system must be able to integrate almost every relevant information source and to optimize the user's information request to the specific requirements perfectly.
- If a user starts a query by just one word an information system should detect all possible topics the user might was searching for. That means that an intelligent information system should collect documents from different sources and create a result set that covers all the different meanings. Moreover an intelligent clustering may help the user to find the information he is searching for quickly and easily. Additionally long text might be automatically summarized to a meaningful abstract[16].

- Existing search engines often apply a fixed strategy for rating results. But if someone is frequently using the information system, the system should learn the individual user's preferences and adapt on the individual needs. This can be done by judging potentially relevant documents not only by one strategy but by a collection of different ratings strategies, some of them optimized on the individual user preferences. The strategy applied for combining the results of different rating strategies should consider the respective scenario and the reliability of each filtering strategy.

3 Approach

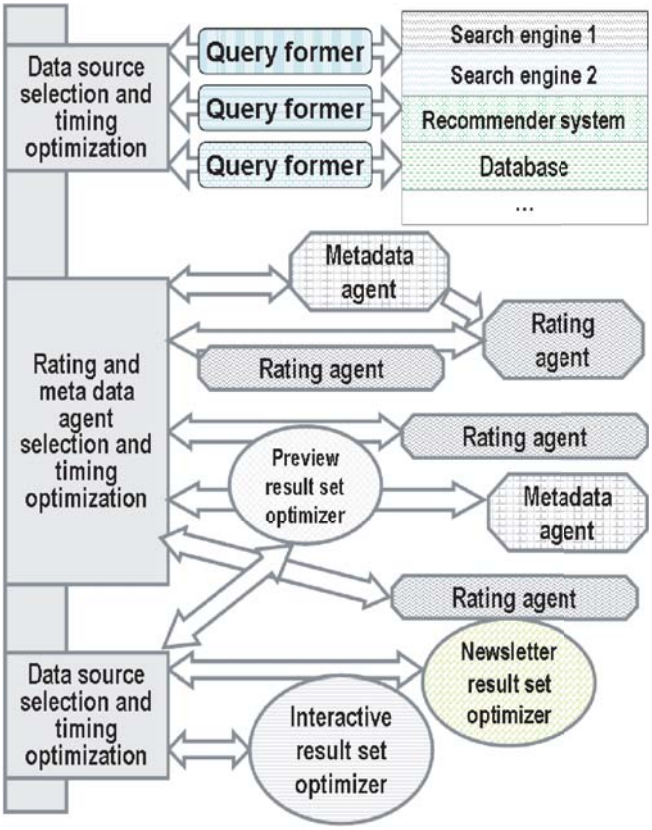

Figure 1: Filter framework architecture

For determining all the relevant data it is usually necessary to analyze potentially relevant documents from a big number of sources, rate each document based on several different strategies and deliver the user only the most relevant

result, taking into account the individual user preferences, the search context and knowledge about sources and other users.

The first layer handles the integration of data sources. If a new query is received by the search system the "Source Manager" analyzes the request and selects the potentially relevant sources. Then the request is forwarded to the selected sources and the potentially relevant documents are retrieved in parallel. For converting and optimizing the queries for the different data source, we define a query former for each source.

The second layer rates the documents retrieved from the data sources and generates additional metadata (especially ratings). Each rating strategy is wrapped by an intelligent agent. The Rating Agent Manager selects for each request the most promising agents considering the dependencies between the agents and determines the required parameter values (e.g. maximal run time) for each agent.

The third layer addresses the optimization of the result set. Based on the ratings calculated in layer II an optimized result set is derived. The used optimization strategy is selected based on the respective search scenario.

Each layer is controlled by a manager agent who monitors the agent performance, their reliability and the demand of resources.

4 Implementation

For proving the suggested approach, we implemented several information systems based on the suggested architecture. There are optimized systems e.g. for filtering scientific documents (PIA+COMM, `http://pia.cs.tu-berlin.de`), for news (Personal Newspaper Service, `http://pzd.dai-labor.de:7780/sky/begin.html`) and for general information ("Personal Information Agent", `http://www.dai-labor.de/pia`). The filtering components are implemented in the programming language Java (`http://java.sun.com/`), the user interface is implemented as web-application using Apache Tomcat (`http://tomcat.apache.org/`). Due to the fact that the filtering framework is based on a service-oriented architecture, some system components may also be implemented in alternative programming language. For instance, we implemented several rating strategies in C because of performance reasons.

The following section describes some of the implemented components of the intelligent information filtering system in detail.

5 The Source Layer

The system is designed to integrate a high number of sources. Each source is controlled by a query forming agent that optimizes the user information request to the special requirements of each source. Each source is treated as an autonomously running intelligent component (agent) that can be easily added

or removed from the system. Currently, agents for following information sources exist:

- Databases, such as CORDIS (`http://cordis.europa.eu`), ACM (`http://www.acm.org`) and IEEE (`http://www.ieee.org`)
- Web portals, such as AgentLink (`http://www.agentlink.org/`) or newspaper portals
- Search engines, such as Yahoo (`http://www.yahoo.com`) or Citeseer (`http://citeseer.ist.psu.edu`).
- Directories from the local file system
- Mailing lists (Integration of POP3- and IMAP-Mail-Servers)
- Recommender systems (based on collaborative rating databases)

Depending on the respective scenario the source manager selects the most relevant information sources. The user's query is converted according to the requirements of each source. This includes a query optimization (e.g. by enhancing the query by scenario specific keywords or by deriving additional constraints on file format or up-to-dateness). Sources especially relevant for a request might be requested several time (based on different parameter settings).

5.1 The Filtering Layer

In the second layer of the system the documents, retrieved as potentially relevant, are ranked by different filtering strategies. The idea of this layer is to have a wide variety of filtering strategies combined to benefit from their specific advantages [2, 19]. The calculated ratings are merged in the third layer to get a reliable and top-quality rating. For instance there are content-based filtering strategies, using indexes with exact and fuzzy matching in the full text and the document's meta data, as well as collaborative filtering algorithms based on analyzing similarities of user ratings [12].

Currently we implemented 25 different strategies for rating potentially relevant documents. Due to the fact that some of the strategies can be deployed with different parameter settings for some of the strategies different agents exist. Some of the rating strategies are:

- A content based rating based on the similarity between the user query and the documents calculated based on tf-idf statistics[1].
- An agent that detects the occurrence of query keywords in the document address (URI).
- To decide whether a document belongs to the domain relevant for the user, we created a decision tree that predicts the domain based on the tf-idf statistic of the respective document.
- For judging how a document matches the individual user preferences we train a Support vector machine (SVM) with the last 100 last document ratings provided by the user.
- An alternative method for predicting how a document matches the user's preferences is to determine the N most similar documents rated by the

[1] tf-idf: **t**erm **f**requency **i**nverse **d**ocument **f**requency

user. Based on the ratings and the calculated similarity the weighted average is calculated. The method can be adapted by considering not only the ratings of the respective user but by taking into account similar users or even all users, to overcome the cold start problem[18].
- A component for detecting unwanted entries (e.g. spam) exists. It is implemented based on a Naive Bayes classifier that can be trained by the user.
- For collaborative filtering we implemented a component that analyzes how the document was rated by users that used similar requests. Based on these ratings the document relevance is predicted.
- For judging scientific papers we defined an agent that analyzes the references between papers and calculates a reputation rating for the respective document and the author.

In addition to the agents calculating the relevance or the quality of documents, additional information about documents can be added auxiliary, applying methods of feature extraction like part-of-speech tagging or semantic analysis.

A "filter manager agent" controls which rating strategies are calculated for each query. The manager agent chooses the most promising agents according to the scenario and the user request based on a predefined rule set. The results, the resource demand as well as the result quality (based on user ratings) are logged and used for learning new rules and improving existing rules. Due to the fact that the filter manager only knows the in- and output of the rating agents, but treats a rating agent as a black box, new rating strategies can be easily added.

5.2 Creating Optimal Filter Result Sets

The third layer of the filtering framework deals with the combination of ratings from different agents (calculated in the filtering layer) and the creation of optimal result sets. For this layer several alternative strategies have been implemented, each showing specific strengths and weaknesses:

- **Fast Preview:** For the very fast creation of result sets a components exists, that combines the first results from the considered information source. It removes duplicates but does not requires rating from filtering components, so that a preview on the filter results can be created immediately after collecting results from the sources.
- **Rule based combination of ratings:** For creating good results sets based on ratings provided by several different filtering strategies, alternative strategies for combining ratings have been implemented. The filtering framework provides components that construct the final filter result set by selecting the documents that have received the highest rating combining the rating from different filtering strategies based on the minimum-, maximal- or sum-function ($Comb^*$ [6]).
- **Optimization using a constraint solver:** A slow but appreciable method for creating filter result set based on documents rated by several strategies is the use of constraint solvers. Therefore we define an objective function that describes an optimal result set. A constraint solver

determines the subset of potentially relevant documents so that the set is optimal according to the objective function.

5.3 Tools for Supporting the User

Besides the filtering component, an intelligent information system should include several functionalities for managing documents and stored requests. We implemented these functionalities optimized on the collaborative work of scientists in the PIA+COMM-system. The filter results are presented similar to e-mail programs where users can sort the results according to several criteria like author or date.

To support the user in finding the most relevant documents, the system provides an automatic identification of clusters of similar documents. Clustering (section 5.4) is especially useful if a search term has different semantic meanings. Users usually are aware of the meaning of their search term and can focus on the cluster they are is interested in most [17, 15, 5].

To help users being up to date in their research area and satisfying their continuous information need, the system provides the functionality of periodically pushing new relevant documents ("alerts"). Users can exactly define the time and the preferred structure of their personal alert. Whether results are presented as a list, as clusters or as alert, it is always possible to rate a document, to improve the quality of future queries and hence to build the PIA+COMM dataset.

Because many users have problems in formulating their information need with adequate queries, the system provides a keyword assistant that supports users in optimizing their queries. For this purpose the keyword assistant analyzes the user queries and selects a promising optimization strategy. The current PIA+COMM system uses optimization strategies based on collaborative filtering, cluster analysis as well as semantic dictionaries (such as WordNet[2]).

5.4 Clustering

Clustering is the process of grouping items in a way that items in a group are similar to each other and dissimilar to the items in other groups [10].

In information retrieval often the problem for retrieving the desired documents given a short query lies in the ambiguity of the query. Terms in a query can occur in various documents and contexts and usually it is very difficult to identify the documents where a search term is used in the intended sense. Therefore, many search results in a result list of a classic information retrieval system are irrelevant. Clustering results for information retrieval is a way to circumvent this problem by structuring the (probably) very long list of search results. Of course, this will not avoid any irrelevant documents but if the cluster labels are intuitive enough, the user can directly choose the cluster which contains the most relevant documents for his request.

[2] `http://wordnet.princeton.edu/`

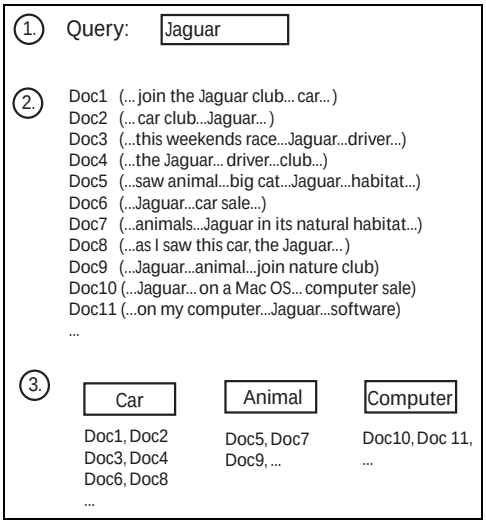

Figure 2: Didactic example of clustering documents for information retrieval

Figure 2 illustrates the process of document clustering. Starting with a user querying an information retrieval system with the phrase "jaguar", a list of relevant documents is found by matching strings in the search phrase with documents. While most search engines stop after the second step (the actual retrieval of documents), text clustering proceeds by analyzing the documents in the result list in order to group similar documents. The user who queried for "jaguar" can then choose among the clusters and access the corresponding documents.

6 Evaluation

The document sources as well as and rating strategies are monitored to measure the quality (by comparing the rating provided by the rating strategies with user ratings and based on benchmark datasets (TREC, `http://trec.nist.gov/data.html`) and the resource demand of each strategy. Additional to the properties of each single strategy also the correlations between the strategies are analyzed. Beside of the automated data collection the user satisfaction and acceptance will be evaluated by interviewing users.

7 Conclusion and Future Work

In this paper we introduced an architecture for an intelligent filtering meta-search system that is open and extensible for new source, rating and meta-data agents as well a result set optimization strategies. In contrast to traditional meta-search engines it integrates a big number of sources of different types and provides a unified rating for all documents. In addition to the often used content-

based ratings we developed several innovative agents judging documents based on user feedback, ratings of similar users. The first evaluation results point out remarkable improvements in the quality and reduced variation range.

In the next steps of the project we will focus on the intelligent combination of heterogeneous filtering strategies. While selecting the most promising filtering agents and combining the results the individual user preferences as well as the search context are considered.

References

1. E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM Press.
2. S. Albayrak and D. Milosevic. Resource and Job Aware Coordination in Multi Agent Filtering Framework. In *Artificial Intelligence and Applications*, pages 539–544, 2005.
3. S. Albayrak and D. Wieczcorek. JIAC—A toolkit for telecommunication applications. In S. Albayrak, editor, *Intelligent Agents for Telecommunication Applications — Proceedings of the Third International Workshop on Intelligent Agents for Telecommunication (IATA'99)*, volume 1699 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag: Heidelberg, Germany, 1999.
4. C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM Press.
5. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, June 1990.
6. E. Fox and J. Shaw. Combination of multiple searches. In *Proceedings of the 2nd Text REtrieval Conference (TREC-2)*, pages 243–252. National Institute of Standards and Technology Special Publication, 500-215, 1994.
7. S. Fricke, K. Bsufka, J. Keiser, T. Schmidt, R. Sesseler, and S. Albayrak. Agent-based telematic services and telecom applications. *Communications of the ACM*, 44(4):43–48, Apr. 2001.
8. M. Hahne, C. Jung, J. Kunegis, A. Lommatzsch, and A. Paus. Ein gradualisiertes communitymodell zur bildung wissenschaftlicher gemeinschaften und seine anwendung auf die entwicklung von social software. In C. M. Norbert Gronau, editor, *Bildung von sozialen Netzwerken in Anwendungen der Social Software*. GITO Verlag – Expertenwissen für die industrielle Praxis, 2007.
9. S. Hochreiter and K. Obermayer. Support Vector Machines for Dyadic Data. *Neural Computation*, pages 1472–1510, 2006.
10. A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Upper Saddle River, NJ, 1988.
11. T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM Press.

12. J. Kunegis, S. Schmidt, and Şahin Albayrak. Modeling Similarity using Electrical Resistance Networks with Negative Edges. 2006.
13. J. H. Lee. Analyses of multiple evidence combination. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–276, New York, NY, USA, 1997. ACM Press.
14. D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining (SDM'05)*, 2005.
15. A. Leuski. Evaluating document clustering for interactive information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 33–40, New York, NY, USA, 2001. ACM Press.
16. I. Mani. *Automatic Summarization*, volume 3 of *Natural Language Processing*. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2001.
17. Mehlitz, Martin. Text Clustering by Selected Latent Semantic Concepts. Master's thesis, Technical University of Berlin, 2006.
18. S. E. Middleton, H. Alani, N. R. Shadbolt, and D. C. De Roure. Exploiting synergy between ontologies and recommender systems. In *In Proceedings of Semantic Web Workshop 2002 at the Eleventh International World WideWeb Conference*. ACM Press, 2002. Reviewed Workshop Proceedings.
19. Y. Rasolofo, F. Abbaci, and J. Savoy. Approaches to collection selection and results merging for distributed information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 191–198, New York, NY, USA, 2001. ACM Press.