Captain Nemo: a Metasearch Engine with Personalized Hierarchical Search Space

Stefanos Souldatos, Theodore Dalamagas and Timos Sellis
School of Electrical and Computer Engineering, National Technical University of Athens, 157 73, Athens, GR
{stef, dalamag, timos}@dblab.ntua.gr and http://www.dblab.ntua.gr/∼ {stef, dalamag, timos}

Keywords: web search, personalization, metasearch engine, classification, hierarchy

Received: November 18, 2005

Personalization of search has gained a lot of publicity the last years. Personalization features in search and metasearch engines are a follow-up to the research done. On the other hand, text categorization methods have been successfully applied to document collections. Specifically, text categorization methods can support the task of classifying Web content in thematic hierarchies. Combining these two research fields, we have created Captain Nemo, *a fully-functional metasearch engine with personalized hierarchical search spaces.* Captain Nemo, *initially presented in [SDS05], retrieves and presents search results according to personalized retrieval models and presentation styles. Here, we present the hierarchical Web page classification approach newly adopted.* Captain Nemo *lets users define a hierarchy of topics of interest. Search results are automatically classified into the hierarchy, exploiting hierarchical k-Nearest Neighbor classification techniques.*

Povzetek:

1 Introduction

Searching for Web content can be extremely hard. Web content can be found in a variety of information sources. The number of these sources keeps increasing, while at the same time sources continually enrich their content. Not only should users identify these sources, but they should also determine those containing the most relevant information to satisfy their information need.

Search and metasearch engines are tools that help the user identify such relevant information. Search engines retrieve Web pages that contain information relevant to a specific subject described with a set of keywords given by the user. Metasearch engines work at a higher level. They retrieve Web pages relevant to a set of keywords, exploiting other already existing search engines.

Personalization on the Web is an issue that has gained a lot of interest lately. Web sites have already started providing services such as preferences for the interface, the layout and the functionality of the applications. Personalization services have also been introduced in Web search and metasearch engines. However, those services deal mostly with the presentation style and ignore issues like the retrieval model, the ranking algorithm and topic preferences.

On the other hand, text classification methods, including k-Nearest Neighbor (k-NN) [Yan94, MLW92], Support Vector Machines (SVM) [Joa98, DPHS98], Naive Bayes (NB) [MN98, BM98], Neural Networks [NGL97], decision trees and regression models, have been succesfully applied to document collections (see [YL99] for a full examination of text classification methods).

Such methods can support the task of classifying Web content in thematic hierarchies. Organizing Web content in thematic categories can be useful in Web search, since it helps users easily identify relevant information while navigating in their personal search space.

There are two main approaches for classifying documents in thematic hierarchies:

– *Flat Model* (Flatten the hierarchy): Every topic of the hierarchy corresponds to a separate category having its own training data. A classifier, based on text categorization techniques determines the right category for a new incoming Web document.

– *Hierarchical Model* (Exploit the hierarchy): A hierarchy of classifiers is built such that every classifier decides each time to classify a document in the appropriate category among the categories of the same level in the hierarchy, following a path from the root down to the leaves of the hierarchy tree. For example, an incoming document might be added to *Arts* category (between *Arts*, *Science* and *Sports*), then to *Dance* category inside *Arts* (between *Poetry*, *Photography* and *Painting*), then to *Spanish_Dances* inside *Arts/Dance*. The assignment scores for all these decisions can determine the final category for the incoming document.

Combining these two research fields, namely personalization of search and Web content hierarchical classification, we have created *Captain Nemo*, a fully-functional metasearch engine with personalized hierarchical search spaces. *Captain Nemo*, initially presented in [SDS05], retrieves and presents search results according to personalized retrieval models and presentation styles. In this paper, we present the hierarchical Web page classification approach, recently adopted in *Captain Nemo*. Users define a hierarchy of topics of interest. Search results are automatically classified into the hierarchy, exploiting Nearest-Neighbour classification techniques.

Our classification approach is a hybrid one. Every topic of the hierarchy is considered to be a separate category having its own training data, as in the flat model. However, the training data set of a topic is enriched by data from its subtopics. As a result, the decision of whether a Web page belongs to a category strongly depends on its descendants.

A typical application scenario for *Captain Nemo* starts with a set of keywords given by the user. *Captain Nemo* exploits several popular Web search engines to retrieve Web pages relevant to those keywords. The resulting pages are presented according to the user-defined presentation style and retrieval model. We note that users can maintain more than one different *sets of preferences*, which result to different presentation styles and retrieval models. For every retrieved Web page, *Captain Nemo* recommends the most relevant topic of user's personal interest. Users can optionally save the retrieved pages to certain folders that correspond to topics of interest for future use.

Contribution. The main contributions of our work are:

– We expand personalization techniques for metasearch engines, initially presented in [SDS05].

– We suggest semi-automatic hierarchical classification techniques in order to recommend relevant topics of interest to classify the retrieved Web pages. The thematic hierarchy is user-defined.

– We present a fully-functional metasearch engine, called *Captain Nemo*[1], that implements the above framework.

Related Work. The need for Web information personalization has been discussed in [SC03] and [SMBR04]. Following this, several Web search and metasearch engines[2] offer personalization services. For example, Alltheweb offers the option to use personal stylesheets to customize the look and feel of its search page. Altavista provides styles to present the retrieved Web pages with high or low detail. The metasearch engines WebCrawler, MetaCrawler, Dogpile can group the Web pages retrieved according to the search engine that actually retrieves them. Regarding the retrieval model, several metasearch engines let the user define the search engines to be used (e.g. Query Server, Profusion, Infogrid, Mamma, Search, Ixquick). Some of them (e.g. Query Server, Profusion, Infogrid, Mamma) have a timeout option (i.e. time to wait for Web pages to be retrieved). Also, Query Server and Profusion offer the option of setting the number of Web pages retrieved by each engine. To the best of our knowledge, there

[1] http://www.dblab.ntua.gr/∼ stef/nemo/

[2] Google, Alltheweb, Yahoo, AltaVista, WebCrawler, MetaCrawler, Dogpile, etc.

is not any metasearch engine that offers the option of setting the weights of the search engines for the ranking of the retrieved pages.

Concerning the topics of interest, [BLP+04] supports that topic-based search will be necessary for the next generation of information retrieval tools. Inquirus2 [GFL+01] uses a classifier to recognize Web pages of a specific category and learn modifications to queries that bias results toward documents in that category. In other words, it biases all results towards the selected category. Northern Light[3] has an approach called *custom folders* that organizes search results into categories. However, these categories are created dynamically by the search results and do not reflect the users' personal interest.

Recently, many researchers have looked into the problem of classifying Web content into thematic hierarchies, using either the flat or the hierarchical model. The former approach has shown poor results, since flat classifiers cannot cope with large amounts of information including many classes and content descriptors. In [LF98], a n-gram classifier was used to classify Web documents in Yahoo categories, based on different combinations of document descriptions (title, summary, category description, category name). [GLF99] presents a probabilistic interpretation of the k-NN classifier to automatically categorize Yahoo Web documents. [FK98] presents a classification method based on a probabilistic inference engine. [CDI98] proposes statistical models for hypertext categorization by exploiting link information in a small neighborhood around documents. The hierarchical approach has been explored initially in [KS97], using a small hierarchy based on a subset of the Reuters-22173 collection (3 top level and 6 second level categories). Experiments with bayesian classification models showed the superiority of the hierarchical model over the flat.

Outline. The rest of this paper is organized as follows. The personalization features of *Captain Nemo* are discussed in Section 2. Section 3 presents the hierarchical classification algorithm that recommends relevant topics of interest to classify retrieved Web pages. The architecture of *Captain Nemo* and several implementation issues are discussed in Section 4. Finally, Section 5 concludes this paper.

[3]http://www.northernlight.com/index.html

2 Personal Search Spaces

Personal search spaces are maintained for users of *Captain Nemo*. Each *personal search space* includes user preferences able to support the available personalization features. In fact, more than one *sets of preferences* can be maintained for each user, which result to different retrieval models and presentation styles. A *personal search space* is implemented through three respective personalization filters, as illustrated in Figure 1.

We next discuss the available personalization features regarding the retrieval model, the presentation style and the topics of interest. The hierarchical Web page classification approach is presented in the following section.

2.1 Personal Retrieval Model

As seen before, most of the existing metasearch engines employ a standard retrieval model. In *Captain Nemo*, this restriction is eliminated and users can create their *personal retrieval model*, by setting certain parameters in the system. These parameters are described below:

Participating Search Engines. Users can declare the search engines they trust, so that only these search engines are used by the metasearch engine.

Search Engine Weights. In a metasearch engine, retrieved Web pages may be ranked according to their ranking in every individual search engine that is exploited. In *Captain Nemo*, as shown in Section 4, the search engines can participate in the ranking algorithm with different weights. These weights are set by the user. A lower weight for a search engine indicates low reliability and importance for that particular engine. The results retrieved by this search engine will appear lower in the output of *Captain Nemo*.

Number of Results. A recent research [iPr04] has shown that the majority of search engine users (81.7%) rarely read beyond the third page of search results. Users can define the number of retrieved Web pages per search engine.

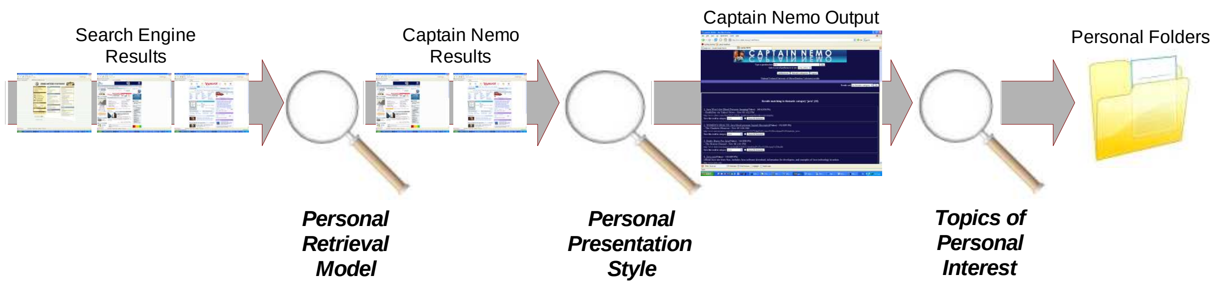


Figure 1: Personal search space.

Search Engine Timeout. Delays in the retrieval task of a search engine can dramatically deteriorate the response time of any metasearch engine that exploits the particular search engine. In *Captain Nemo*, users can set a timeout option, i.e. time to wait for Web pages to be retrieved for each search engine. Results from delaying search engines are ignored.

2.2 Personal Presentation Style.

Captain Nemo results are presented through a customizable interface, called *personal presentation style*. The following options exist:

Grouping. In a typical metasearch engine, the results returned by search engines are merged, ranked and presented in a common list. Beside this typical presentation style, *Captain Nemo* can group the retrieved Web pages (a) by search engine or (b) by topic of interest, as shown in Figure 2. The latter is based on a semi-automatic hierarchical classification technique, which is described in Section 4.

Content. The results retrieved by *Captain Nemo* include three parts, title, description and URL. Users can declare which of these parts should be displayed.

Look and Feel. Users can customize the general look and feel of *Captain Nemo*. Selecting among the available color themes and page layouts, they can define preferable ways of presenting results.

Figure 3 presents the available options for the *personal retrieval model* and *personal presentation style*.

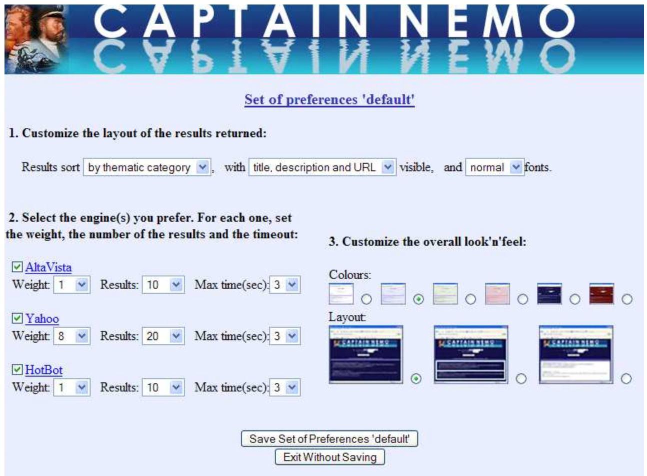


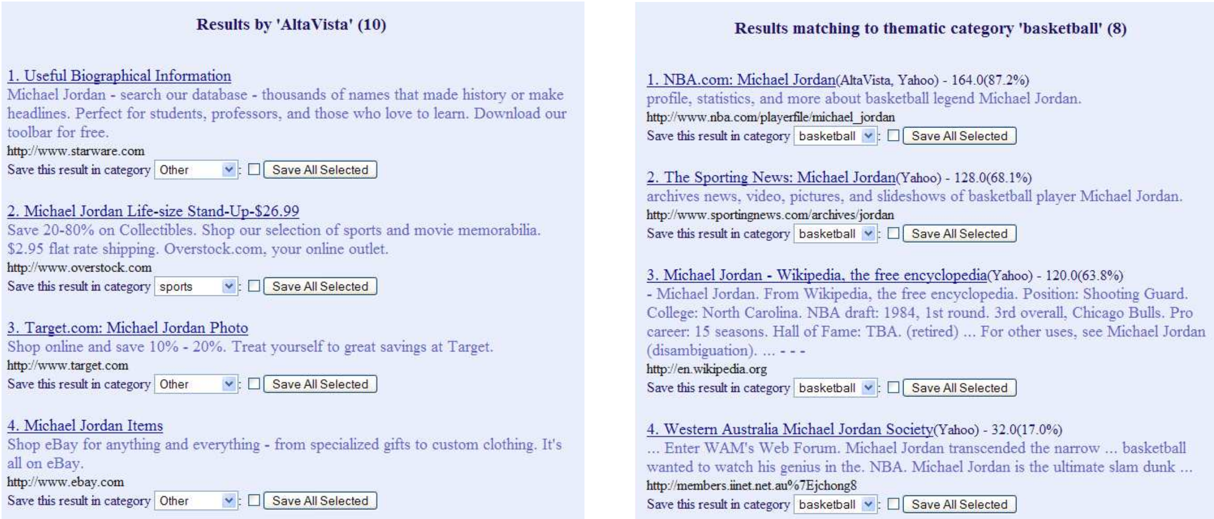
Figure 3: Personalization options.

2.3 Topics of Personal Interest

Captain Nemo users can define *topics of personal interest*, i.e. thematic categories where search results can be kept for future reference. The retrieved Web pages can be saved in folders that correspond to these topics. These folders have a role similar to *Favorites* or *Bookmarks* in Web browsers.

For every retrieved Web page, *Captain Nemo* recommends the most relevant *topic of personal interest*. Users can optionally save the retrieved pages to the recommended or other folder for future use. The hierarchical classification technique, discussed in Section 3, uses a set of keywords describing each *topic of personal interest*.

The *topics of personal interest* are organized in a hierarchy. The hierarchy can be thought of as a tree structure having a *root* and a set of *nodes* which refer to topics of the thematic hierarchy. For every topic node, there is



(a) by search engine (b) by topic of interest

Figure 2: Result grouping alternatives.

– a label that describes its concept and

– a stricter description of the concept.

Figure 4 shows such a hierarchy of *topics of personal interest*. The administration of the same hierarchy in *Captain Nemo* is shown in Figure 5.

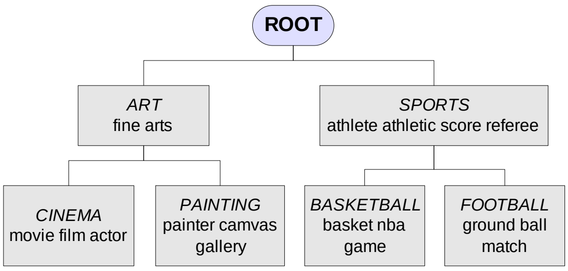


Figure 4: Hierarchy of topics of personal interest.

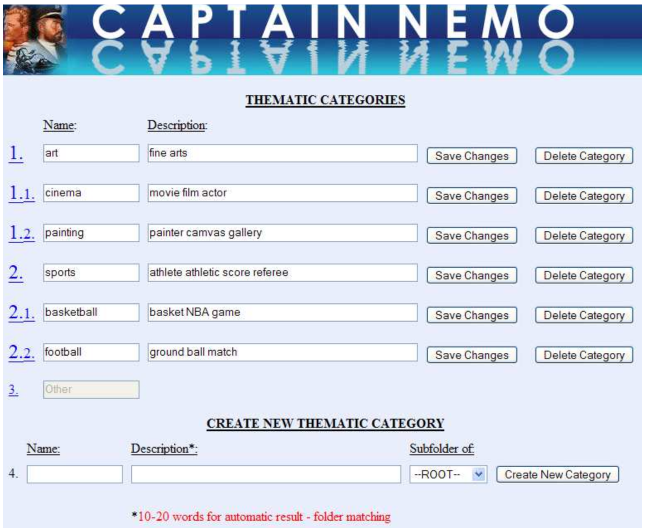


Figure 5: Topics of personal interest.

3 Hierarchical Classification of Retrieved Web Pages

As we have already mentioned, *Captain Nemo* recommends relevant topics of interest to classify the retrieved pages, exploiting k-Nearest Neighbor classification techniques. These topics are organized in a thematic hierarchy. Our classification approach is a hybrid one. Every topic of the hierarchy is considered to be a separate category having its own training data, as in the flat model. However, the training data set of a topic is enriched by data from its subtopics. For example, the hierarchy of Figure 4 is translated to the hierarchy of Figure 6. As a result, the decision of whether a Web page belongs to a category strongly depends on its descendants.

A k-Nearest Neighbor classification method is implemented, since it was shown to be effective [YL99]. Instead of training documents, *Captain Nemo* uses the topic descriptions set by the user. The description of a retrieved Web page in-

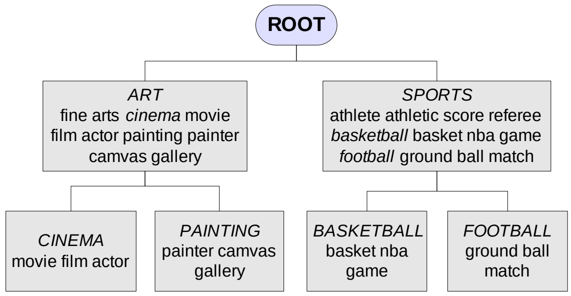


Figure 6: Enriched hierarchy.

cludes its title and a part of its content (extracted by search engines). The classification algorithm identifies the most relevant topic of interest for all retrieved pages.

k-NN Classification. The k-NN classification method presumes that a group of categories is defined for a data set and a set of training documents corresponds to each category. Given an incoming document, the method ranks all training documents according to the similarity value between those documents and the incoming document. Then, the method uses the categories of the k top-ranked documents to decide the right category for the incoming document by adding the per-neighbour similarity values for each one of those categories [Yan94, YL99]:

$$y(\mathbf{x}, c_j) = \sum_{\mathbf{d}_i \epsilon kNN} sim(\mathbf{x}, \mathbf{d}_i) \times y(\mathbf{d}_i, c_j) \qquad (1)$$

where:

1. $\mathbf{x}$ is an incoming document, $\mathbf{d}_i$ is a training document, c_j is a category,
2. $y(\mathbf{d}_i, c_j) = 1$ if $\mathbf{d}_i$ belongs to c_j or 0 otherwise,
3. $sim(\mathbf{x}, \mathbf{d}_i)$ is the similarity value between the incoming document $\mathbf{x}$ and the training document $\mathbf{d}_i$,

Using thresholds on these scores, k-NN obtains binary category assignments and allows the system to assign a document to more than one categories. Instead it can just use the category with the highest score as the right one for the incoming document. *Captain Nemo* follows the second approach.

To be more specific, *Captain Nemo* needs to calculate similarity measures between the description of each retrieved Web page and the description of every *topic of personal interest*. The similarity measure employed is a $tf - idf$ one [WMB99]. Let D be the description of a topic of interest and R the description of a retrieved Web page. The similarity between the topic of interest and the retrieved Web page, $sim(R, D)$, is defined as follows:

$$Sim(R, D) = \frac{\sum_{t \epsilon R \cap D} w_{R,t} \times w_{D,t}}{\sqrt{\sum_{t \epsilon R \cap D} w_{R,t}^2} \times \sqrt{\sum_{t \epsilon R \cap D} w_{D,t}^2}} \qquad (2)$$

where t is a term, $w_{R,t}$ and $w_{D,t}$ are the weights of term t in R and D respectively. These weights are:

$$w_{R,t} = \log\left(1 + \frac{C}{C_t}\right) \qquad (3)$$

$$w_{D,t} = 1 + \log f_{D,t} \qquad (4)$$

where C is the total number of topics of interest, C_t is the number of topics of interest including term t in their description and $f_{D,t}$ is the frequency of occurence of t in description D.

Having a new, retrieved Web page, we rank the topics of interest according to their similarity with the page (the topic of interest with the highest similarity will be on the top). Then, the top-ranked topic of interest is selected as the most appropriate for the retrieved page.

Example. We have created a user with the hierarchy of *topics of personal interest* presented in Figure 4. For this user, we have run the query "michael jordan", asking for just a few results. A screenshot of the results grouped by topic of interest is shown in Figure 2(b). Totally, there are:

– 0 results in category *1. ART*,

– 3 results in category *1.1. CINEMA*,

– 2 results in category *1.2. PAINTING*,

– 3 results in category *2. SPORTS*,

– 8 results in category *2.1. BASKETBALL*,

– 0 results in category *2.2. FOOTBALL*.

As expected, the majority of results are matched to topic *BASKETBALL*. However, there are results matching to other topics as well. Results matching to topic *CINEMA* deal with Michael Jordan as an actor. Results in topic *PAINTING* refer to photo galleries with photos of Michael Jordan. Finally, results matching to topic *SPORTS* refer to the athletic background of Michael Jordan in general.

4 System Implementation

This section presents the architecture of our application and discusses various interesting implementation issues. Figure 7 illustrates the main modules of *Captain Nemo*.

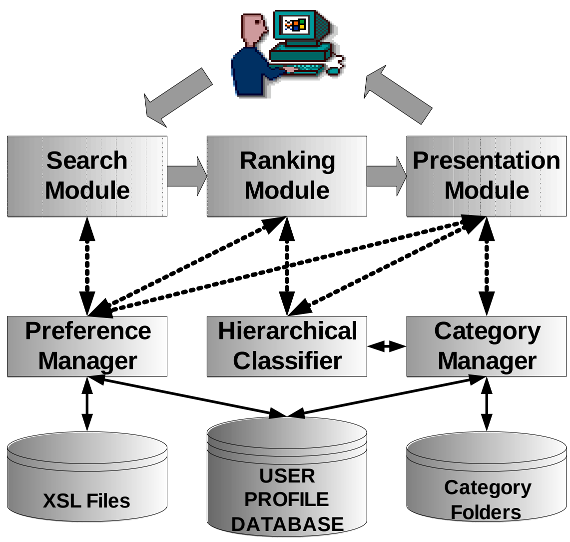


Figure 7: System architecture.

Search Module. It implements the main functionality of the metasearch engine, providing connection to the search engines selected by the user. It retrieves the relevant Web pages according to the retrieval model defined by the user. The results are sent to the ranking module for further processing.

Ranking Module. The retrieved Web pages are ranked and grouped according to the personal retrieval model of the user. For every retrieved Web page, a corresponding topic of personal interest is determined.

Presentation Module. It maintains several XSL filters that construct the resulting pages (wrapped as XML documents) according to the look and feel settings determined by the user.

Preference Manager. It provides the connection between the three aforementioned modules (i.e. search module, ranking module, presentation module) and the information stored in user profiles. It is also responsible for updating user profiles and the used XSL files.

Hierarchical Classifier. It implements the hierarchical classification of results to the thematic hierarchy of the user, as described in Section 3.

Category Manager. It manages the topics of interests and keeps the appropriate folders on disk in accordance with the user profiles. It provides all the necessary information to the hierarchical classifier. It cooperates with the presentation module, when grouping by topics of interest is selected by the user.

Our application is implemented on top of the PostgreSQL database system[4], exploiting Perl CGI scripts to wrap the results of search engines[5].

The next subsection discusses in detail the ranking mechanisms used in our application.

4.1 Ranking

Given a query, a typical metasearch engine sends it to several search engines, ranks the retrieved Web pages and merges them in a single list. After the merge, the most relevant retrieved pages should be on top. There are two approaches used to implement such a ranking task. The first one assumes that the initial scores assigned to the retrieved pages by each one of the search engines are known. The other one does not presupposes any information about these scores.

In [RAS01], it is pointed out that the scale used in the similarity measure in several search engines may be different. Therefore, normalization is required in order to achieve a common measure of comparison. Moreover, the reliability of each search engine must be incorporated in the ranking

[4] http://www.postgresql.org/

[5] http://search.cpan.org/dist/WWW-Search/lib/WWW/Search.pm

algorithm through a weight factor. This factor is calculated separately during each search. Search engines that return more Web pages should receive higher weight. This is due of the perception that the number of relevant Web pages retrieved is proportional to the total number of Web pages retrieved as relevant for all search engines exploited by the metasearch engine.

On the other hand, [Dum94, GP98, TVGJL95] stress that the scores of various search engines are not compatible and comparable even when normalized. For example, [TVGJL95] notes that the same document receives different scores in various search engines and [Dum94] concludes that the score depends on the document collection used by a search engine. In addition, [GP98] points out that the comparison is not feasible not even among engines using the same ranking algorithm and claims that search engines should provide statistical elements together with the results.

In [AM01], ranking algorithms are proposed which completely ignore the scores assigned by the search engines to the retrieved Web pages: bayes-fuse uses probabilistic theory to calculate the probability of a result to be relevant to the query, while borda-fuse is based on democratic voting. The latter considers that each search engine gives votes in the results it returns, giving N votes in the first result, $N-1$ in the second, etc. The metasearch engine gathers the votes for the retrieved Web pages from all search engines and the ranking is determined democratically by summing up the votes.

Weighted Borda-Fuse. The algorithm adopted by *Captain Nemo* is the weighted alternative of Borda-fuse. In this algorithm, search engines are not treated equally, but their votes are considered with weights depending on the reliability of each search engine. These weights are set by the users in their profiles. Thus, the votes that the i result of the j search engine receives are:

$$V(r_{i,j}) = w_j * (max_k(r_k) - i + 1) \tag{5}$$

where w_j is the weight of the j search engine and r_k is the number of results rendered by search engine k. Retrieved pages that appear in more than one search engines receive the sum of their votes.

Example. A user has defined the personal retrieval model of table 1.

Search Engine	Results	Weight	Timeout
SE1	20	7	6
SE2	30	10	8
SE3	10	5	4

Table 1: Personal retrieval model

The user runs a query and gets 4, 3 and 5 results respectively from the three search engines specified. According to Weighted Borda-Fuse, the search engines have given votes to the results. The first result of each search engine receives 5 votes, as the largest number of results returned is 5. Table 2 shows the votes received by the search engines.

Search Engine	1st	2nd	3rd	4th	5th
SE1	5	4	3	2	-
SE2	5	4	3	-	-
SE3	5	4	3	2	1

Table 2: Result votes by search engines

Captain Nemo multiplies these votes by the weight of each search engine in order to push upward results of search engines trusted most by user. The final votes of each result of each search engine is shown in table 3.

Search Engine	1st	2nd	3rd	4th	5th
SE1	35	28	21	14	-
SE2	*50*	40	30	-	-
SE3	25	20	15	10	5

Table 3: Result votes by *Captain Nemo*

So, the first result to appear in the rank is the first result of search engine SE2.

5 Conclusion

Getting this idea from two research fields, namely personalization of search and Web content classification, we have created *Captain Nemo*, a fully-functional metasearch engine with personalized hierarchical search spaces. *Captain Nemo*, initially presented in [SDS05], retrieves and presents search results according to personalized retrieval models and presentation styles. In this paper, we presented the hierarchical Web page classification approach, recently adopted in *Captain Nemo*. Users define a hierarchy of topics of interest. Search results are automatically classified into the hierarchy, exploiting k-Nearest Neighbor classification techniques.

For future work, we are going to improve the hierarchical classification process, exploiting background knowledge in the form of ontologies [BH04]. Next, we will incorporate a Word Sense Disambiguation (WSD) technique in the spirit of [MTV+05]. Finally, a community of users can evaluate the overall functionality of *Captain Nemo*.

References

[AM01] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th ACM SIGIR Conference*, pages 276–284. ACM Press, 2001.

[BH04] S. Bloehdorn and A. Hotho. Text classification by boosting weak learners based on terms and concepts. In *Proceedings of the 4th ICDM Conference*, pages 331–334, 2004.

[BLP+04] W. L. Buntine, J. Löfström, J. Perkiö, S. Perttu, V. Poroshin, T. Silander, H. Tirri, A. J. Tuominen, and V. H. Tuulos. A scalable topic-based open source search engine. In *Proceedings of the ACM WI Conference*, pages 228–234. ACM Press, 2004.

[BM98] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21st ACM SIGIR Conference*, pages 96–103. ACM Press, 1998.

[CDI98] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In Laura M. Haas and Ashutosh Tiwary, editors, *Proceedings of the 17th ACM SIGMOD Conference*, pages 307–318. ACM Press, 1998.

[DPHS98] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and represenations for text categorization. In *Proceedings of the 7th ACM CIKM Conference*, pages 148–155. ACM Press, 1998.

[Dum94] S. T. Dumais. Latent semantic indexing (lsi) and trec-2. In *Proceedings of the 2nd TREC Conference*, 1994.

[FK98] N. Fuhr and C.-P. Klas. A new effective approach for categorizing web documents. In *Proceedings of the 22nd IRSG Conference*, 1998.

[GFL+01] E. Glover, G. Flake, S. Lawrence, W. P. Birmingham, A. Kruger, C. Lee Giles, and D. Pennock. Improving category specific web search by learning query modifications. In *Proceedings of the SAINT Symposium*, pages 23–31. IEEE Computer Society, January 8–12 2001.

[GLF99] N. Govert, M. Lalmas, and N. Fuhr. A probabilistic description-oriented approach for categorizing web documents. In *Proceedings of the 8th ACM CIKM Conference*, pages 475–482. ACM Press, 1999.

[GP98] L. Gravano and Y. Papakonstantinou. Mediating and metasearching on the internet. *IEEE Data Engineering Bulletin*, 21(2), 1998.

[iPr04] iProspect. iProspect search engine user attitudes. http://www.iprospect.com/premiumPDFs /iProspectSurveyComplete.pdf, 2004.

[Joa98] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th ECML Conference*, 1998.

[KS97] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the 14th ICML Conference*, 1997.

[LF98] Y. Labrou and T. Finin. Yahoo! as an ontology - using yahoo! categories to describe documents. In *Proceedings of the 7th ACM CIKM Conference*, pages 180–187. ACM Press, 1998.

[MLW92] B. Masand, G. Linoff, and D. Waltz. Classifying news stories using memory based reasoning. In *Proceedings of the 15th ACM SIGIR Conference*, pages 59–65. ACM Press, 1992.

[MN98] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the Learning for Text Categorization Workshop, at the 15th AAAI Conference*, 1998.

[MTV+05] D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word sense disambiguation for exploiting hierarchical thesauri in text classification. In *Proceedings of the 16th ECML/9th PKDD Conference*, pages 181–192, 2005.

[NGL97] H.-T. Ng, W.-B. Goh, and K.-L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th ACM SIGIR Conference*, pages 67–73. ACM Press, 1997.

[RAS01] Y. Rasolofo, F. Abbaci, and J. Savoy. Approaches to collection selection and results merging for distributed information retrieval. In *Proceedings of the 10th ACM CIMK Conference*. ACM Press, 2001.

[SC03] C. Shahabi and Y-S. Chen. Web information personalization: Challenges and approaches. In *Proceedings of the 3rd DNIS Workshop*, 2003.

[SDS05] S. Souldatos, T. Dalamagas, and T. Sellis. Sailing the web with captain nemo: a personalized metasearch engine. In *Proceedings of the Learning in Web Search Workshop, at the 22nd ICML Conference*, 2005.

[SMBR04] M. Sahami, V. O. Mittal, S. Baluja, and H. A. Rowley. The happy searcher: Challenges in web information retrieval. In *Proceedings of the 8th PRICAI Conference*, pages 3–12, 2004.

[TVGJL95] G. Towell, E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies for information retrieval. In *Proceedings of the 12th ICML Conference*, 1995.

[WMB99] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, 2nd edition, 1999.

[Yan94] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17th ACM SIGIR Conference*, pages 13–22. ACM Press, 1994.

[YL99] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd ACM SIGIR Conference*, pages 42–49. ACM Press, 1999.