Incorporating Conceptual Matching in Search

Juan M. Madrid
EECS Department
University of Kansas
Lawrence, KS 66045
jmadrid@ku.edu

Susan Gauch
EECS Department
University of Kansas
Lawrence, KS 66045
sgauch@ittc.ku.edu

ABSTRACT

As the number of available Web pages grows, users experience increasing difficulty finding documents relevant to their interests. One of the underlying reasons for this is that most search engines find matches based on keywords, regardless of their meanings. To provide the user with more useful information, we need a system that includes information about the conceptual frame of the queries as well as its keywords. This is the goal of KeyConcept, a search engine that retrieves documents based on a combination of keyword and conceptual matching. Documents are automatically classified to determine the concepts to which they belong. Query concepts are determined automatically from a small description of the query or explicitly entered by the user. This paper describes the system architecture, the training of the classifier, and the results of our experiments evaluating system performance. KeyConcept is shown to significantly improve search result precision through its use of conceptual retrieval.

KEYWORDS

Conceptual search, text classification, ontologies.

1. INTRODUCTION

The Web has experienced continuous growth since its creation. As of March 2002, the largest search engine contained approximately 968 million indexed pages in its database [SES 02]. Finding the right information from such a large collection is extremely difficult. One of the main reasons for obtaining poor search results is that many words have multiple meanings [Krovetz 92]. For instance, two people searching for "wildcats" may be looking for two completely different things (wild animals and sports teams), yet they will get exactly the same results. The underlying difficulty is most search engines are based on a string matching algorithm that returns a match whenever an exact occurrence of the search term is found, regardless of its meaning.

To address this problem, we are developing KeyConcept, a search engine that, in addition to the *keywords*, takes into account the topic or *concepts* related to the query. Documents are indexed by their keywords and by concepts automatically chosen from the Open Directory ontology. During retrieval, in addition to the keywords, this search engine accepts one or more concepts to restrict the search domain. The query topics can either be specified manually by the user, or they can found by running a small description of the query through a classifier. This paper reports on experiments run to evaluate the relative importance given to concept matches and keyword matches by the retrieval process.

The reminder of this paper begins with a presentation of previous related work in Section 2. Then, the architecture of KeyConcept is described in Section 3. Descriptions of the experiments concerning the classifier training and system performance follow in Section 4, along with the discussion of the results. We conclude with some observations and ideas for future work on Section 5.

2. RELATED WORK

2.1. Text classification

Text classification organizes information by associating a document with the best possible concept(s) from a predefined set of concepts. Several methods for text classification have been developed, each with different approaches for comparing the new documents to the reference set. These including comparison between vector representations of the documents (Support Vector Machines, k-Nearest Neighbor, Linear Least-Squares Fit, TF-IDF), use of the joint probabilities of words being in the same document (Naïve Bayesian), decision trees and neural networks. A thorough survey and comparison of such methods is presented in [Yang 99], [Pazzani 96] and [Ruiz 99].

As presented by [Chekuri 97] and [Matsuda 99], one of the main uses of text classification is to restrict the search domain. In [Chekuri 97], users have the option of specifying some concepts of interest when submitting a query. Then, the system only searches for results in the specified concepts. The approach in [Matsuda 99] extends this idea by classifying documents based on other attributes, e.g., size, number of images, presence or absence of certain tags, as well as content. In this system, the user has the option of specifying the type of document he/she is looking for, e.g., a catalog, a call for papers, a FAQ, a glossary, etc., in addition to the search terms.

In the OBIWAN project [Pretschner 99], ontologies are used to represent user profiles. Queries are submitted to a traditional search engine, and the results are classified into the ontology concepts based on their summaries. The documents in the result set are re-ranked based on matches between the summary concepts and the highly-weighted concepts in the user profile. While this approach was able to improve rank ordering of the results, it was not able to find more information for users because it could only work on the result set retrieved by a generic search engine. KeyConcept takes this approach one step further by integrating the conceptual matching into the retrieval process itself.

2.2. Ontologies

An ontology is an arrangement of concepts that represents a view of the world [Chaffee 2000] that can be used to structure information. Ontologies can be built by specifying the semantic relationships between the terms in a lexicon. One example of such ontology is Sensus [Knight 94], a taxonomy featuring over 70,000 nodes. The OntoSeek system [Guarino 99] uses this ontology for information retrieval from product catalogs.

Ontologies can also be derived from hierarchical collections of documents, such as Yahoo! [YHO 02] and the Open Directory Project [ODP 02]. [Labrou 99] reports the use of the TellTale [Pearce 97] classifier to map new documents into the Yahoo! ontology. This is done by training the classifier with documents for each one of the concepts in the ontology, and then finding the concept that has a greater similarity with the new document. Furthermore, an ontology can be used to allow users to navigate and search the Web using their own hierarchy of concepts. The OBIWAN project [Chaffee 00, Zhu 99] accomplishes this by letting users define their own hierarchy of concepts, then mapping this personal ontology to a reference ontology (Lycos in this case).

Authors can also embed ontological information in their documents. SHOE (Simple HTML Ontology Extensions) [Heflin 2000] is a language that serves this purpose, allowing the creation of new custom-made ontologies and the extension of existing ones.

3. SYSTEM ARCHITECTURE

During indexing, KeyConcept includes information about the concepts to which each document is related. To accomplish this, the traditional structure of an inverted file was extended to include mappings between concepts and documents. The retrieval process takes advantage of this enhanced index, supporting queries that use only words, only concepts, or a combination of the two. The user can select the relative importance of the criteria (word match or concept match).

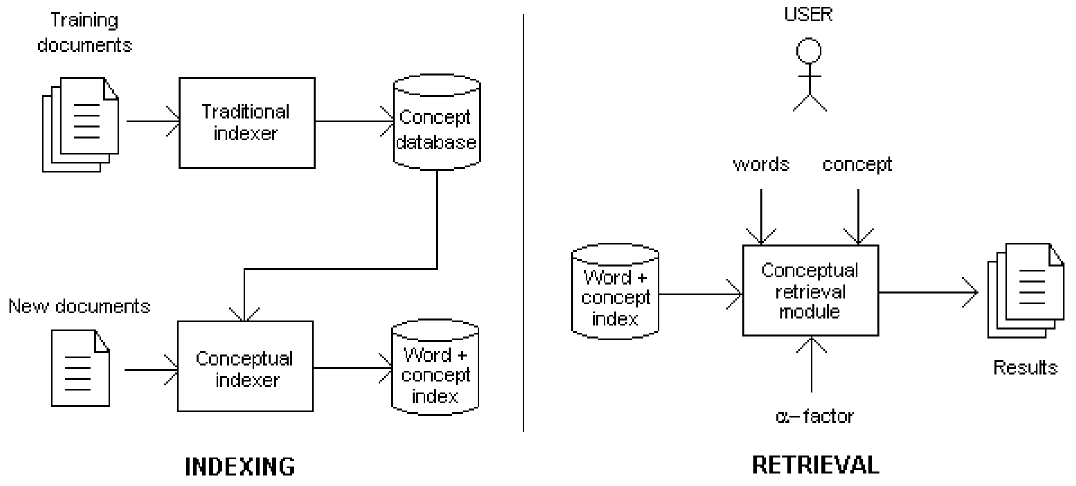

Figure 1. Operation of the conceptual search engine

Indexing

The indexing process is comprised of two phases: Classifier training and collection indexing. During classifier training, a fixed number of sample documents for each concept are collected and merged, and the resulting super-documents are preprocessed and indexed using the *tf * idf* method. This essentially represents each concept by the centroid of the training set for that concept. During collection indexing, new documents are indexed using a vector-space method to create a traditional word-based index. Then, the document is classified by comparing the document vector to the centroid for each concept. The similarity values thus calculated are stored in the concept-based index.

Retrieval

During retrieval, the user provides a query containing words, concept identifiers, or both. For this initial version of KeyConcept, the user also provides a factor (α-factor) between 0 and 1, specifying the relative importance of concept matches to word matches. If α is 1, only concept matches are considered. If it is 0, only word matches matter. When α is 0.5, concept and word matches contribute equally. In future, we expect that the system will have a default value α so that end-users need not enter it. However, we allow it to be adjusted in this version for evaluation purposes. After receiving user data, the search engine performs the search and stores the results for word and concept matches in separate accumulators. The final document scores are computed using the formula:

$$Document\ score = (\alpha \times concept\ score) + ((1-\alpha) \times word\ score)$$

4. EXPERIMENT DESCRIPTION AND RESULTS

Two series of experiments were conducted to evaluate and tune our conceptual search engine. The goal of the first series of experiments was to determine the size of the collection needed to train the classifier, specifically, the number of concepts and the number of training documents per concept. The second series of experiments was designed to determine how much contribution concept-based matching should have on the retrieval process.

Choosing the training data set

Because it is readily available for download from their web site in a compact format, we chose to use the Open Directory Project hierarchy [ODP 02] as our source for training data,. In addition, it is also becoming a widely used, informal standard. As of April 2002, the Open Directory had more than 378,000 concepts. Because of the large volume of data involved, training the classifier would be a long and difficult task if the full set of concepts was used. In addition, subtle differences between certain concepts may be apparent to a human but indistinguishable to a classification algorithm. Thus, we decided to use the concepts in the top three levels of the Open Directory. This initial training set was composed of 2,991 concepts and approximately 125,000 documents.

4.1 Classifier training experiments

To determine how many documents were needed to train each concept, we designed three experiments:

Experiment 1: Determining an upper bound for the number of documents. We wished to test the hypothesis that, beyond a certain number of training documents per concept, classifier precision does not improve and it might even decrease. In this experiment, we trained the classifier using the concepts that had at least 70 associated documents (633 of the 2,991 concepts). From those documents, two were reserved for testing and the rest were used for training. Since the classifier returned the top ten concept matches for each test case, this allowed us to compute the precision for exact matches (when the true concept for the document comes up first in the classifier output), and for matches between the top two and the top ten (when the true concept appears in places two to ten).

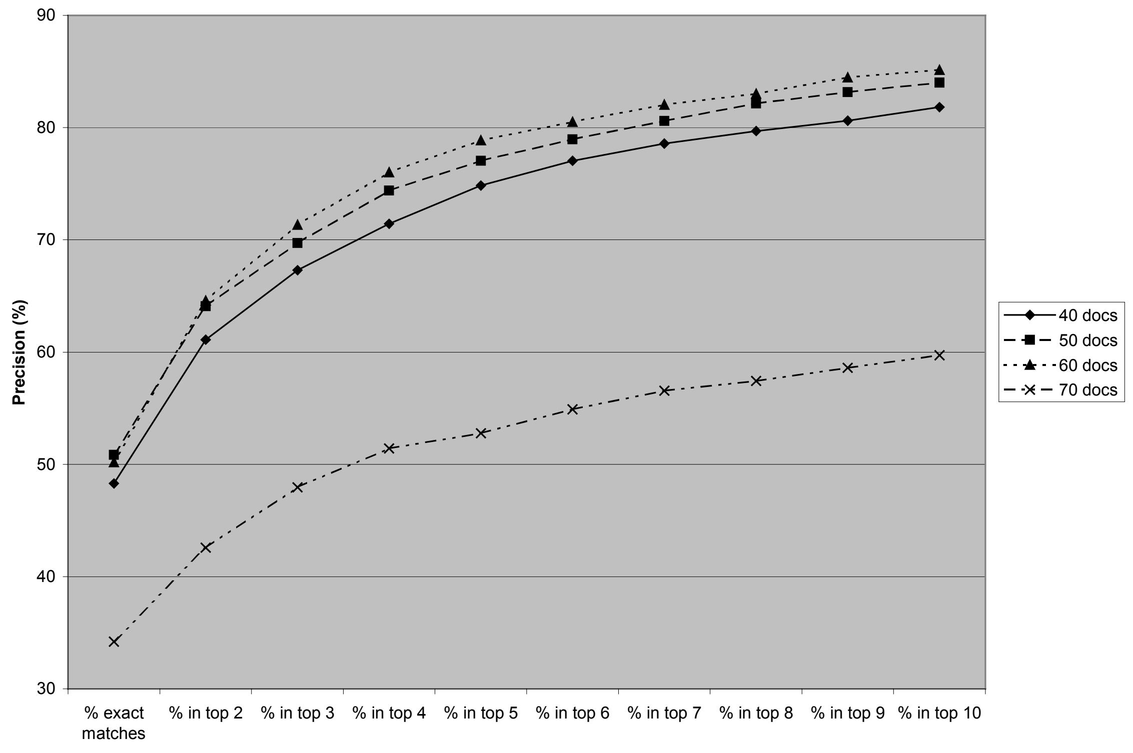

Chart 1. Determination of upper bound for number of documents per concept

For this experiment, the classifier was trained four times, starting with 40 training documents per concept on the first run, and adding 10 training documents in each subsequent run. As shown in Figure 1, with 40 documents, the classifier achieves an exact match precision of 48.2%. For 50 documents, we reach a peak of 50.9%. Then, the exact match score drops to 50.2% with 60 training documents, although the top 2 to top 10 scores still experience a small increase. Finally, the exact match score drops dramatically to 34.2% when all 70 training documents are used. Thus, our hypothesis proves to be valid, and the upper bound for the number of training documents per concept is 60.

Experiment 2: Determining a lower bound for the number of documents. The purpose of this experiment is to determine the minimum number of documents per concept needed to train the classifier. Since many concepts from the ODP hierarchy have fewer than 60 documents, if we can get acceptable precision with fewer documents, we can increase the number of concepts used from the ODP hierarchy. The same procedure of Experiment 1 is used, but this time we decreased the number of training documents per concept and measured the effect in precision. Note that the number of concepts involved here is greater than in Experiment 1 since we used all concepts having at least 50 training documents (941 of the 2,991 concepts).

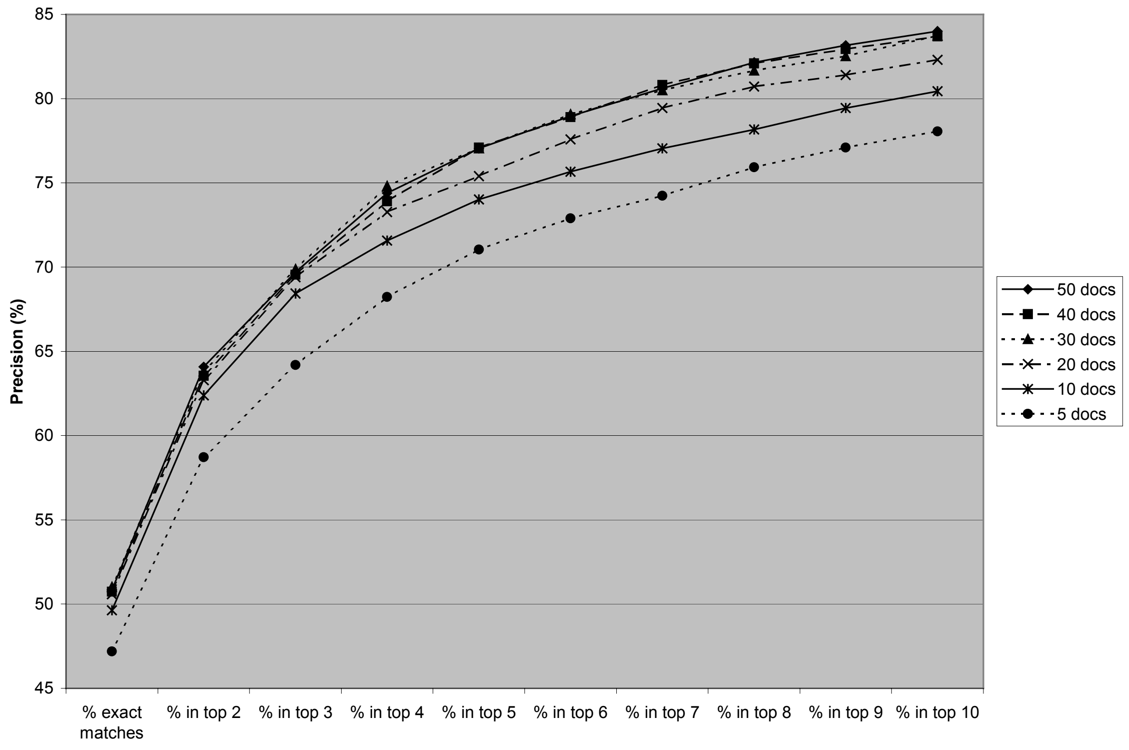

Chart 2. Determination of lower bound for number of documents per concept

In this experiment, we trained the classifier six times, starting with 50 training documents per concept and using ten fewer documents in each subsequent run, except for the last run in which we used only 5 documents per concept. As shown in Figure 2, the precision values for 50, 40 and 30 documents are almost coincident. For 20 documents, we observe a decrease in the precision starting with the precision in the top 5, and for 10 documents the entire precision curve is shifted down. This leads us to conclude that 30 documents is a lower bound for the number of training documents per concept. We analyzed the data and found that using more than 30 documents per concept does not increase the classifier precision by a significant value (p-value of chi-square test ≈ 1). Thus, we decided to use 30 documents per concept to train the classifier in all the following experiments. This allowed us to use a final training database consisting of 46,920 documents from the 1,564 concepts that have at least 30 documents. With 30 training documents, we achieve an accuracy of 51% for exact matches of documents to concepts, 77.05% for matches in the top 5 results and 84% in the top 10.

Experiment 3: Relationship between number of concepts and precision (Scalability). This experiment tries to determine whether or not the precision of the classifier is independent of the number of concepts between which the classifier must decide. Also, dependency on the particular set of concepts chosen is evaluated. To accomplish this, the classifier is trained on different subsets of the 1,564 concepts having 30 training documents. First, 100 concepts are chosen at random, the classifier is trained on their documents, their test cases are submitted to the classifier, and the percentage of exact concept matches is determined. Next, another 100 concepts are chosen at random, the classifier is trained on 200 concepts, and all 200 test cases are classified. The process is repeated until all concepts have been processed. The experiment is then repeated several times so that different subsets are chosen. If the classifier precision is independent of the number of concepts, the precision should remain constant as the number of concepts increases. Also, if the classifier precision is indeed independent from the particular concept sets used in each run, the precision of exact concept matches should not vary between runs.

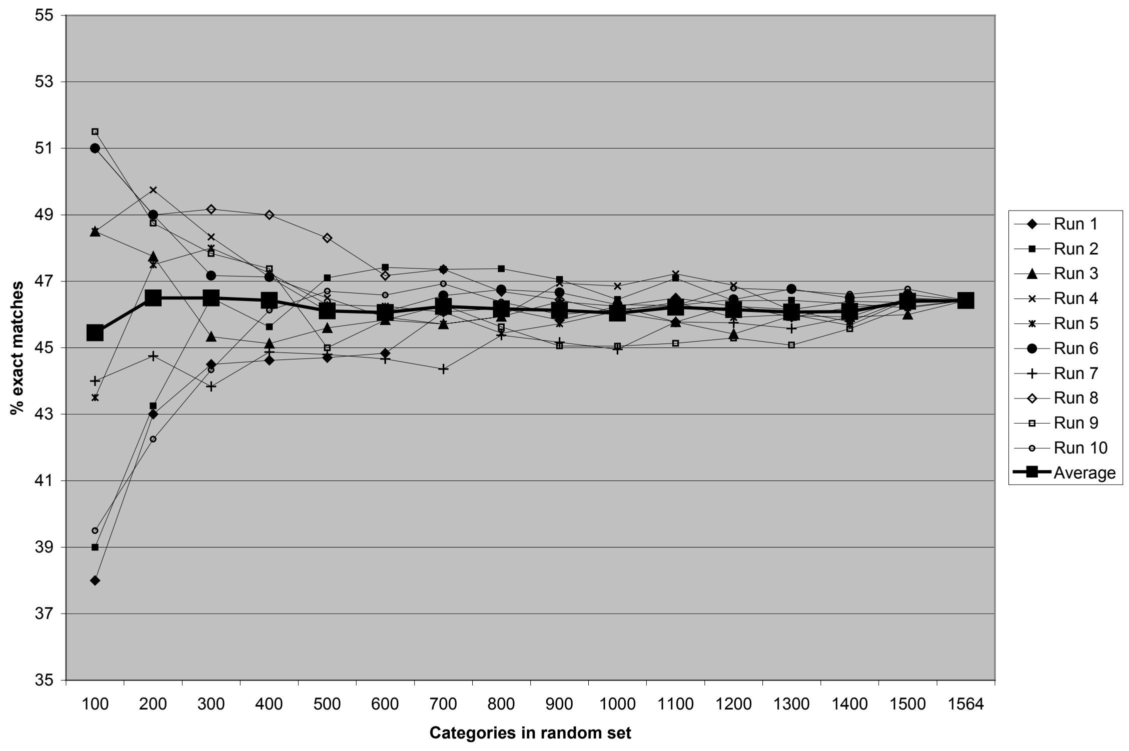

Chart 3. Relationship between number of concepts and precision

After training our classifier using the final training database from last experiment, we created 10 random sequences of concepts, ran experiment 3 for each one of them, and averaged the results. Chart 3 shows results for the 10 runs, and the average score in a darker line. The average precision score is approximately constant at 46%, showing that the accuracy of the classifier does not degrade as the number of concepts increases. In addition, when only 100 concepts are used, the precision varies from 38% to 51%. However, when there are at least 600 concepts, all runs are between 45% and 47%, so the particular concepts chosen do not affect the results for large enough concept sets. This illustrates the independence between the number of concepts and the classifier precision.

4.1.1. Discussion

The experimental results show that using 30 training documents per concept allows us to reach a good compromise between the amount of training data and the classifier precision. If more documents per concept are used, fewer concepts would meet the amount of training data requirement, thus decreasing the specificity of the classification.

On the other hand, using too few documents for training each concept makes each concept too specific and inadequately represented by its training data. Hence, as the number of concepts increases, the differences between them become too subtle for an automatic classifier to distinguish between them. Finally, the classifier accuracy does not deteriorate as the number of concepts is increased.

4.2. Retrieval precision experiments

After training of the classifier, we designed two experiments to study whether or not a search engine that combines conceptual and keyword matching outperforms a search engine based only on keywords. For this purpose, we built a test set consisting of 100,000 documents chosen from the WT2g collection [Hawking 99]. The test set includes the 2,279 documents having positive relevance judgments and 97,721 randomly selected documents. Because the title section of each of the 50 WT2g's topics resembles a typical search engine query (2 to 3 words in length) [Zien 01], we used the title section as the keyword query for our search engine, Then, we ran the description and narrative paragraphs included with each topic through our classifier and used the obtained concepts as the concept input for the search engine. These descriptions and titles average 50 words in length.

It is worth noting we also manually determined the best matching concepts for each topic and input the manually chosen concepts along with the query. However, since the difference between the results obtained using the automatically obtained concepts and the manually obtained ones is not significant (p-value of chi-square test $\approx$ 1), it is not discussed further.

Experiment 4: Effect of α on search precision

In this experiment, we tried to find a good value for α, the balance between concept matches and keyword matches. We varied two factors in this experiment: (1) the α factor, and (2) the number of top concepts for each query that are actually used for retrieval. Then, we calculate the top-10 and top-20 precision score by counting the number of relevant results in the top 10 and 20 results of each query, and averaging the scores across all queries.

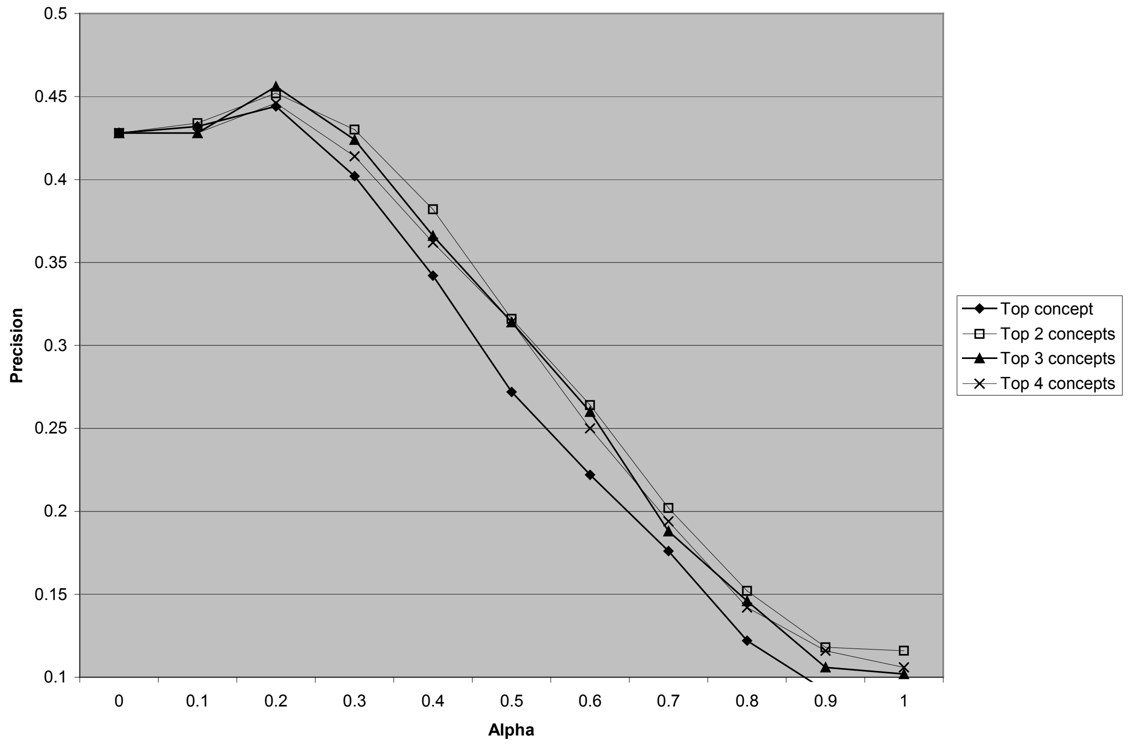

Chart 4. Effect of α factor on top 10 precision

The baseline for this experiment is when α=0, i.e., only keyword matching is used. In this case, the precision is 42.8% for the top 10 case and 33.5% for the top 20 case.

As shown in Figure 4, the best results for top 10 precision (44.4%) are obtained when the top 3 concepts for each query are used as the concept input, for α=0.2. Precision scores drop sharply for α values greater than 0.2, meaning that the conceptual score alone is not enough to achieve a good precision in searches and, not surprisingly, the keywords are the more important factor in identifying relevant documents. Precision also drops slightly when more than 3 concepts are used for retrieval, meaning additional concepts actually introduce noise in the search by boosting the score of non-relevant documents.

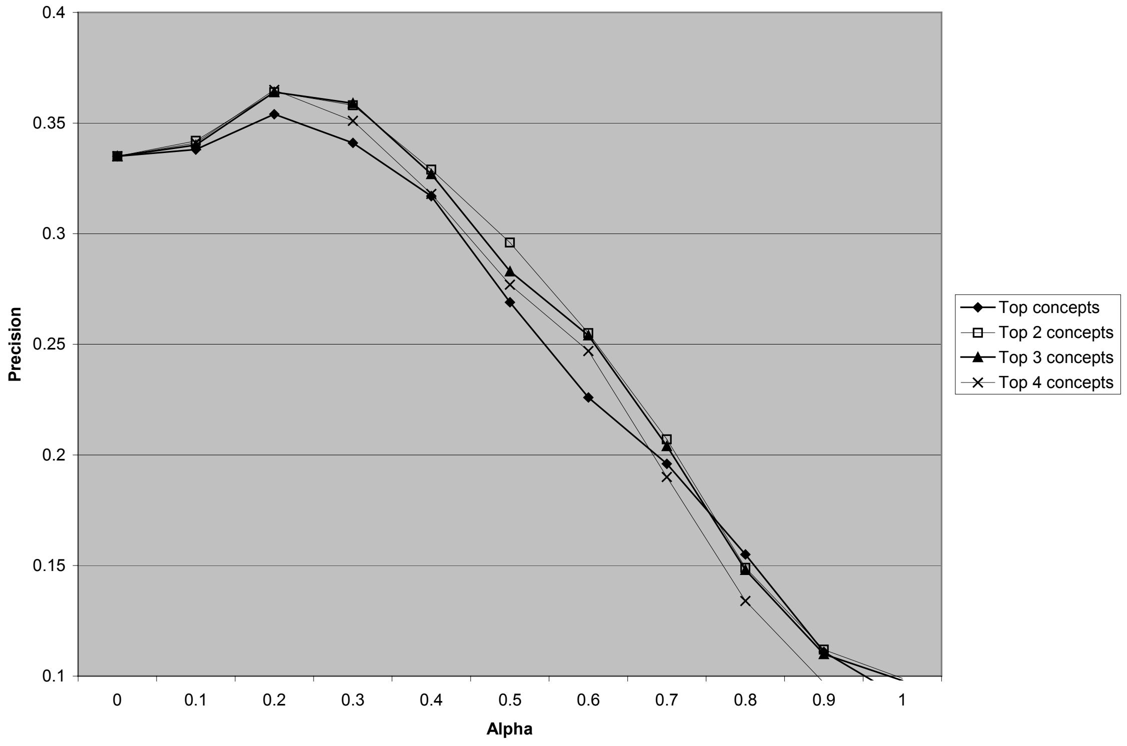

Chart 5. Effect of α factor in top 20 precision.

Figure 5 shows the experiment results for top 20 precision. The precision score also reaches a peak (at 36.5%) for α=0.2, when using the top 3 concepts of each query as input.

Experiment 5: Per-query precision score comparison

The purpose of this experiment is to compare the performance of each of the WT2g topics when run through the engine with α=0 (word match only) and α=0.2 (best word + concept match). In the searches including concept matching, the top 3 concepts for each query were used as the query input for our engine.

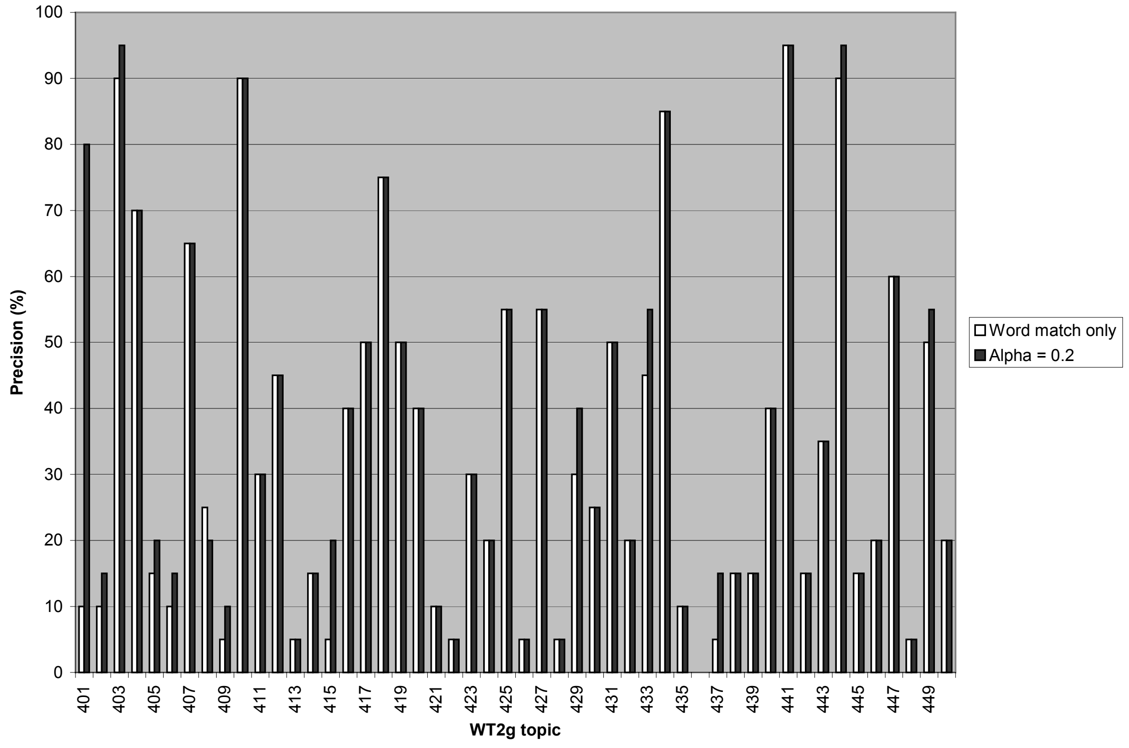

Chart 6. Per-query top-20 precision score comparison

Results of this experiment are shown on chart 6. From the 50 queries available in the collection, 12 experienced an increase in precision when using combined keyword and concept retrieval, 1 experienced a decrease, and 37 showed no variation. Although small, the difference between keyword-only and keyword-concept retrieval performance is significant (p-value of chi-square test ≈ 10^{-6}). For queries where no new relevant results were found, the use of combined keyword and conceptual retrieval did not have a significant effect on the ranking of the results.

4.2.1. Discussion

We found a value of 0.2 for the α factor increases the precision of the proposed searches. This indicates that the keyword matching is far more important than concept matching, but that search precision can be significantly increased when concept matching is included. The concepts for a given query can be determined automatically from associated descriptive text, and performance is best when the top three concepts thus identified are used during the matching process. The need for more than just the top matching concept is likely due to the fact that that, as shown in Experiment 2, the correct concept is only the top classifier result 50% of the time, but it appears in the top 3 concepts 70% of the time.

We believe the reason for the rather modest precision increases obtained in our experiments resides in the characteristics of the WT2g queries. Definition of relevance for these queries is very restrictive. A document having both keyword and concept matches could be deemed as non relevant by some semantic or content detail obvious to a human evaluator, but which is undetectable for an automatic classifier. This causes the number of relevant documents per query to be very small, on average 40, and many documents that are on topic are judged irrelevant.

5. CONCLUSIONS AND FUTURE WORK

This paper presented the idea of a conceptual search engine, where information about the concepts related to each document is stored in the index. When querying the engine, the conceptual information is used in addition to the keywords to provide results that are more relevant to the user.

This paper showed it is possible to improve search precision when conceptual information about the documents is used in addition to the keywords for retrieval. We used a subset of the Open Directory hierarchy to build our concept database, and determined that using 1,564 concepts from level 3 of the hierarchy, with 30 training documents per concept, was adequate. In the retrieval experiments, we obtained a top-20 precision of 36.5%, which was a significant improvement over keyword search alone. We intend to perform more experiments using broader, user-based relevance judgments to see if a further increase in precision and recall is possible.

Currently, the concepts for each query must be provided manually or by running a text related to the query through a classifier. Most users see this as a burdensome task. Thus, providing the query concepts in a completely automatic fashion would be beneficial. In the near future, we plan to provide the concepts by means of user profiling technology.

The engine would compare the query to a user profile and extract the best matching concepts from the profile. This way, users will receive results more suited to their interests and current tasks.

REFERENCES

[Chaffee 2000] Jason Chaffee, Susan Gauch. Personal Ontologies For Web Navigation. In Proceedings of the 9th International Conference On Information Knowledge Management (CIKM), 2000, pp. 227-234.

[Chekuri 97] Chandra Chekuri, Michael H. Goldwasser, Prabhakar Raghavan, Eli Upfal. Web Search Using Automatic Classification. In Proceedings of the 6th International WWW Conference. Santa Clara (CA), USA, 1997. http://www.scope.gmd.de/info/www6/posters/725/web_search.html

[Guarino 99] N. Guarino, C. Masolo, and G. Vetere, OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems, 14(3), May 1999, pp. 70-80.

[Hawking 99] David Hawking, Ellen Voorhees, Nick Craswell, and Peter Bailey. Overview of the TREC8 Web Track. In Eighth Text REtrieval Conference (TREC-7), November 1999. http://citeseer.nj.nec.com/article/hawking99overview.html

[Heflin 2000] Jeff Heflin, James Hendler. Dynamic Ontologies on the Web. In Proceedings of 17th National Conference on Artificial Intelligence (AAAI), 2000. http://www.cs.umd.edu/projects/plus/SHOE/pubs/aaai2000.pdf

[Knight 94] K. Knight and S.K. Luk. Building a Large-Scale Knowledge Base for Machine Translation. In Proceedings of the 12th National Conference on Artificial Intelligence (AAAI), 1994, volume 1, pp. 773-778.

[Krovetz 92] Robert Krovetz and Bruce W. Croft. Lexical Ambiguity and Information Retrieval. ACM Transactions on Information Systems, 10(2), April 1992, pp. 115-141.

[Labrou 99] Yannis Labrou, Tim Finin. Yahoo! As An Ontology – Using Yahoo! Categories To Describe Documents. In Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), 1999, pp. 180-187.

[Matsuda 99] Katsushi Matsuda, Toshikazu Fukushima. Task-Oriented World Wide Web Retrieval By Document Type Classification. In Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), 1999, pp. 109-113.

[ODP 02] Open Directory Project. http://dmoz.org

[Pazzani 96] Michael Pazzani, Jack Muramatsu, Daniel Billsus. Syskill & Webert: Identifying Interesting Web Sites. In Proceedings of the 13th National Conference On Artificial Intelligence, 1996, pp. 54-61.

[Pearce 97] Claudia Pearce, Ethan Miller. The TellTale dynamic hypertext environment: Approaches to scalability. In Advances in Intelligent Hypertext, Lecture Notes in Computer Science. Springer-Verlag, 1997.

[Pretschner 99] Alexander Pretschner, Susan Gauch. Ontology Based Personalized Search. In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), November 1999, pp. 391-398.

[Ruiz 99] Miguel Ruiz, Padmini Srinivasan. Hierarchical Neural Networks For Text Categorization. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 1999, pp. 281-282.

[SES 02] Search Engine Showdown: Size statistics. http://www.searchengineshowdown.com/stats/sizeest.shtml

[Yang 99] Yiming Yang, Xin Liu. A Re-Examination Of Text Categorization Methods. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 1999, pp. 42-49.

[YHO 02] Yahoo! http://www.yahoo.com

[Zhu 99] Xiaolan Zhu, Susan Gauch, Lutz Gerhard, Nicholas Kral, Alexander Pretschner. Ontology-Based Web Site Mapping For Information Exploration. In Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), 1999, pp. 188-194.

[Zien 01] Jason Zien, Jörg Meyer, John Tomlin, Joy Liu. Web Query Characteristics And Their Implications On Search Engines. In Proceedings of the 10th International WWW Conference. Hong Kong, China, 2001. http://www10.org.hk/cdrom/posters/1077.pdf