

KeyConcept: Conceptual Search and Pruning Exploiting Concept Relationships

Susan Gauch¹, Devanand Ravindran² and Aravind Chandramouli³

¹*Department of Computer Science & Computer Engineering, University of Arkansas, Fayetteville, AR 72704;* ²*Department of Electrical Engineering & Computer Science, University of Kansas, Lawrence, KS, 66044;* ³*Computerized Assessment and Learning, 2512 West 6th Street, Suite A, Lawrence, KS 66049; U.S.A.*

ABSTRACT

As the number of available Web pages grows, users experience increasing difficulty in 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 better serves user information needs by augmenting queries with information about the user's conceptual framework. This is the goal of KeyConcept, a conceptual search engine. During indexing, KeyConcept classifies documents into concepts selected from a manually-constructed concept hierarchy. During retrieval, KeyConcept ranks documents based on a combination of keyword and conceptual similarity. This paper describes the system architecture and discusses the results of experiments that demonstrate significant improvements in search effectiveness using our approach to conceptual retrieval. Specifically, we compare the precision obtained for a test query set. In addition, when conceptual relationships are used to prune search results, we also achieve significant increases in search result precisions. Finally, when the two approaches, conceptual retrieval and conceptual pruning, are used in combination, we achieve a statistically significant improvement from 36.7% for traditional keyword based search to 63.77% for conceptual search.

KEYWORDS: conceptual search, intelligent search, Web page ranking

Address for correspondence: Susan Gauch, Department of Computer Science & Computer Engineering, University of Arkansas, Fayetteville, AR 72704, USA; e-mail: sgauch@uark.edu; aravind81@gmail.com

1. INTRODUCTION

The Web has experienced continuous growth since its creation. A study in 2005 estimates the size of the indexable Web to be more than 11 billion pages (Gulli & Signorini 2005). Finding the right information from such a large collection is difficult. One of the main reasons for obtaining poor search results is that many words have multiple meanings (Krovetz & Croft 1992). 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.

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 (<http://www.ittc.ku.edu/keyconcept/>). Documents are indexed by their keywords and by concepts selected from the Open Directory (ODP 2005) concept hierarchy. The classifier is *trained* on the documents linked to the appropriate concepts and then, during indexing, documents are classified into the top matching concepts. During retrieval, in addition to the keywords, KeyConcept accepts one or more concepts that restrict the search domain. Because most users, over 72%, never look beyond the first page of search results (Jansen & Spink, 2006), we evaluated our new system by looking at the number of relevant documents in the first page of the results (i.e., the precision in the top 10 results, or P₁₀).

Our initial evaluation shows that, when compared with keyword-only matching, conceptual retrieval significantly improves precision. We found, however, that requiring an exact match between the user-supplied concept and those identified during training limited our ability to improve search results. We then present revised algorithms that exploit the concept relationships in the ODP concept hierarchy as part of document ranking and to prune out irrelevant results. Although exploiting related concepts does not improve document ranking, using them to prune irrelevant results after keyword-based retrieval does significantly improve the precision of the result set. Finally, we combine conceptual retrieval with conceptual pruning, producing the best results overall.

The incorporation of conceptual matching in search engines is an important first step in providing personalized search. In future, rather than having the users explicitly provide their concepts of interest, these concepts could be inferred by the search engine based on user actions and stored in a user profile. Thus, the concepts that accompany the keyword query could be extracted from the user profile, or user’s

current context, reducing the burden on the user and also avoiding the problem of user-error during concept selection.

2. RELATED WORK

2.1 Text Classification

Text classification organizes information by associating a document with the best matching concept(s) from a set of concepts. Classification requires a predefined set of concepts, also called classes or categories, and information describing what types of documents belong in each concept. In general, this knowledge takes the form of a set of documents that have been manually classified into each concept. Classification usually occurs in two phases: the *training* phase in which the classifier learns which features best represent each concept, and the *classification* phase during which new, unclassified documents are placed into the best matching concepts. During training, features are extracted from the training documents and used to represent the concept. During classification, features are extracted from the new document and compared with the concept features to identify the best matches.

There has been a tremendous amount of research into classification in general, and text classification in particular. The various approaches differ in how the concepts and documents are represented, how the features are extracted and weighted, and how the similarity between the documents and concepts is calculated. One simple approach is the Rocchio classification (Rocchio 1971) in which each concept is represented by a single vector, created from the training documents. During classification, each new document is classified into the most similar concept. This similarity is typically determined by calculating the cosine similarity between the document vector and each concept's representative vector. Other well-known classifiers are the *k*-Nearest Neighbor (*k*NN) (Dasarathy 1991), Naïve Bayes (NB) (Ferguson 1973) and Support Vector Machine (SVM) classifiers (Vapnik 2000).

In the KeyConcept system, the classification is done during indexing. We compared the effectiveness and efficiency of Rocchio, *k*NN, Naïve Bayes, and SVM classifiers for our collection and found that the Rocchio algorithm provided the highest accuracy (Gauch 2006). This result was somewhat surprising because SVM generally provides the highest accuracy (Dumais & Chen 2000, Joachims 1998),

although the inappropriateness of SVM for very large feature/concept spaces has been pointed out in literature (Dhillon et al. 2002). As described in Section 3.1, we calculate the term weights for the concept vectors so that terms occurring in multiple training documents for the concept are weighted higher.

2.2 Ontologies

The concept hierarchies used in this work are a simple version of more general knowledge structures called ontologies. An ontology is an arrangement of concepts that represents a view of the world that can be used to structure information (Chaffee & Gauch 2000). Ontologies can be built by specifying the semantic relationships between the terms in a lexicon. One example of such ontology is Sensus (Knight & Luk 1994), a taxonomy featuring over 70,000 nodes. The OntoSeek system (Guarino et al. 1999) uses this ontology for information retrieval from product catalogs. Ontologies can also be derived from hierarchical collections of documents, such as Yahoo! (Yahoo 2005) and the Open Directory Project (ODP 2005). Labrou and Finin (1999) report on the use of the TellTale classifier (Pearce & Miller 1997) to map new documents into the Yahoo! ontology. Rather than using the concept hierarchy to narrow the scope of a query, Sieg et al. (2003) use Yahoo! concept hierarchy to expand the set of query terms and then perform textual retrieval. Each expanded term is assigned a term weight that is pre-calculated using conceptual relevance obtained from the hierarchy.

Liu et al. (2002) and Pitkow et al. (2002) have devised systems that use personalization to identify relevant concepts as related to the individual user. Liu et al. approach the problem of personalizing information retrieval according to user preferences by analyzing previous queries made by the user and developing a user profile. The user profile is then used to select the most appropriate concepts for a user when a user makes a new query. Pitkow et al. use *contextualization* to obtain relevant search results from an Internet search engine. The ODP structure is used here to identify the user's search context. The user's past browsing history is also stored. This information is used to augment a query by adding relevant terms from the user's profile and current context. An ontology can also be used to provide personalized browsing of the Web. The OBIWAN project (Chaffee & Gauch 2000, Zhu et al. 1999) 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 explicitly embed ontological information in their documents. SHOE (Simple HTML Ontology Extensions) (Heflin & Hendler 2000) is a language that serves this purpose, allowing the creation of new custom-made ontologies and the extension of existing ones. This is the approach used by the semantic Web, yet it may be a long time before most authors explicitly annotate their content with concepts selected from a universal ontology (Tirri 2003).

2.3 Conceptual Search

Early approaches to conceptual retrieval focused on expanding queries from an online thesaurus or knowledge base (Gauch et al. 1999). However, because words are ambiguous, these approaches are likely to add noise when they expand the incorrect word sense. In contrast, current conceptual approaches represent the documents contents at a higher level of abstraction. Allan et al. (2003) define contextual retrieval as the combination of search technologies and knowledge about the query and user context to provide the most appropriate answer for a user's information needs. Although a variety of approaches for gathering information about the user are being developed, there must be similar efforts in extraction of conceptual information from documents so that the user's context and the documents can be matched on the conceptual level.

The last few years have seen an increased focus on creating concept-based search tools. In contrast with traditional keyword search, for conceptual search concepts must be given along with the query. These concepts can either be automatically inferred or users can explicitly specify them. Klink et al. (2002) define the concept based on maintaining a history of query term(s). Chang et al. (2006) try to infer users' interest by constructing query concepts from the document space and comparing it with the initial query and choosing the most similar concepts for the query. In contrast, users have the option of specifying some concepts of interest when submitting a query in (Chekuri et al. 1997; Zhang et al. 1999). Then, the system only searches for results in the specified concepts.

Irrespective of how the concepts are determined, conceptual matching must be performed using the query and concept. Conceptual matching can be performed by either having it as part of indexing and retrieval for a search engine or by building conceptual search over a keyword-based search engine. Aggarwal and Yu (2001) propose a method of conceptual indexing of documents by identifying common

word-chains that occur within a concept. In a more recent work, Baziz et al. (2005) use semantic networks for conceptual indexing of documents. They have shown that using conceptual indexing along with traditional keyword indexing improves the precision when compared to either of these approaches separately. In contrast to these techniques that have indexing as part of the search engine, in the OBIWAN project (Pretschner & Gauch 1999, Gauch et al. 2004a), conceptual matching is performed on the results obtained from a traditional search engine as part of personalization. User profiles are represented by weighted concept hierarchies where the weight of a concept represents the user's interest in that topic. The concept weights are determined implicitly by classifying web pages browsed by the user. 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 then re-ranked based on matches between the summary concepts and the highly weighted concepts in the user's 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. This work extends the previous work by incorporating conceptual match into the document selection process rather than just reranking documents selected via keyword matches alone.

Another conceptual search project, the Semantic Information Retrieval System (Cesarano et al. 2003) incorporates conceptual search over a keyword based search engine. The user submits a query containing keywords and sentences, and a depth level to which each site is to be searched. The prototype is a combination search/spidering system that searches, downloads results, classifies them to detect semantic matches, and then explores promising links automatically. Although the system's precision seems quite high on the sample queries, no formal evaluation of the system's effectiveness is presented and efficiency is a concern with an interactive system. Another example of a system that provides conceptual matching over regular search is the Yahoo mindset (Yahoo, 2006). Users provide a query and the results are re-organized depending on whether user's search is commercial or non-commercial. This information is provided using a slider over the search results interface.

Our research is similar to (Chekuri et al. 1997, Zhang et al. 1999, Yahoo, 2006) in that users submit the concepts in which they are interested. Similar to Baziz et al. (2005), we have conceptual matching as part of the indexing and retrieval. However,

efficiency is a concern with their approach and their results are on a very small collection with 25 concepts and 7,823 documents while we present results on a collection with approximately 125,000 documents and 2,991 concepts. In addition, we also present results obtained by varying the importance of the keyword and concept matches. Also, we have not come across any work that exploits the hierarchical conceptual structure along with the concept provided by the user to improve the search results by pruning unwanted results.

3. SYSTEM ARCHITECTURE

During indexing, KeyConcept automatically identifies and stores the concepts to which each document is related. The retrieval process uses this enhanced index, supporting queries that use only keywords, only concepts, or a combination of the two. For our experimental system, we provide a mechanism to vary the relative importance of the keyword and concept matches using an input parameter, the α factor.

3.1 Indexing

The indexing process is comprised of two phases: classifier training and collection indexing. During classifier training, a fixed number of sample documents

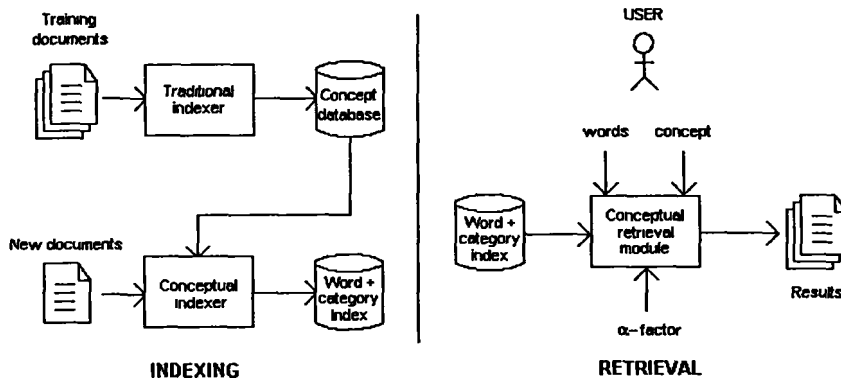


Fig. 1: KeyConcept Architecture

for each concept are collected and used as examples for the concept. For efficiency, we employ a Rocchio classifier in which each concept is represented by a single vector of weighted terms, c_m . The terms are extracted from the training set and weighted using a variation of the vector space term weighting formula that incorporates information about how many training documents contain the term. Essentially, the traditional $tf * idf$ formula is expanded by including two idf factors instead of the usual one factor: the idf calculated between concepts (icf) and the idf calculated between documents in a given concept's training set (cdf). As shown in Eq. (1), the weight of term i in concept m is calculated as:

$$wt_{im} = tf_{im} * icf_i * cdf_{im} \quad (1)$$

where

tf_{im} = the total frequency of term i in all training documents for concept m

icf_i = the inverse concept frequency

$$= \log \frac{N}{cf_i}$$

where

N = Total Number of Concepts

cf_i = Number of Concepts containing term i

cdf_{im} = the concept document frequency

$$= \log \frac{df_{mi}}{N_m}$$

where

df_{mi} = Number of Training Documents for Concept m containing term i

N_m = Total Number of Training Documents in Concept m

Thus, we weight a term based on its total frequency in the training set for a concept, tf_{im} , the rarity of the term in the training sets for other concepts, icf_i , and the rarity of the term within the training set for that concept, cdf_{im} . Since icf_i is equivalent to the traditional idf_i calculated per concept rather than per document, these factors are essentially the traditional $tf * idf$ weight applied to concepts with the addition of the cdf_{im} , which is essentially the idf calculated between documents in a given concept's training set. This formula was compared with a variety of other weighting schemes in (Gauch, 2004b) and found to be statistically significantly more effective than $tf * idf$ alone and other combinations of these three factors.

The dimensionality of the concept vectors is very large, one dimension for every word used in any document in the collection. This dimensionality is somewhat reduced by removing stop words and further reduced by stemming. However, since the training set for each concept contains only a small fraction of the possible words, and absent terms receive a weight of 0.0, these concept vectors are very sparse. Thus, for fast access during classification, our classifier stores the mappings from terms to concept weights calculated during training in a traditional inverted file.

During indexing, we create a pair of vectors for each document. The first is a traditional document-keyword vector, and the second is a document-concept vector that stores the weight of every concept for the document. In the keyword vector for document j , dk_j , the weight of term i in document j , wt_{ij} , is calculated using

$$wt_{ij} = tf_{ij} * idf_i \quad (2)$$

where tf_{ij} = the total frequency of term i in document j

idf_i = the inverse document frequency

$$= \log \frac{N}{df_i}$$

where N = Total number of training documents

df_i = Number of documents containing term i

For the document-concept vector for document j , dc_j , the weight of concept m in document j , wt_{mj} , is computed by calculating the similarity between the document-keyword vector dk_j and the concept vector ck_m created during training using the cosine similarity measure, as shown in equation 3.

$$wt_{mj} = \text{sim}(dk_j, ck_m) = \frac{\bar{dk}_j \cdot \bar{ck}_m}{|\bar{dk}_j| \times |\bar{ck}_m|} \quad (3)$$

$$= \frac{\sum_{i=1}^t wt_{ij} \times wt_{im}}{\sqrt{\sum_{i=1}^t wt_{ij}^2} \times \sqrt{\sum_{i=1}^t wt_{im}^2}}$$

where t is the number of terms in the vocabulary

3.2 Conceptual Retrieval

During retrieval, the user provides a query containing words and concept identifiers. Similar to the document representation, queries are represented as a pair of vectors, a keyword vector, qk , and a concept vector, qc . After receiving the user input, the search engine performs a traditional vector-space keyword search in which the documents are weighted based on the similarity between the document keyword vectors and the query keyword vectors using Eq. (5). The results of this search, the normalized keyword scores, are stored in a keyword accumulator. In addition, KeyConcept also performs a conceptual search in which the documents are weighted based on the similarity between the document concept vectors and the query concept vectors using equation 6. The results of this search, the normalized concept scores, are stored in a concept accumulator. Equation (4) shows the calculation of the final document score for document d given query q using a weighted combination of the weights of the documents in the two accumulators. A parameter, α , controls the relative contributions of the keyword vector similarity and the concept vector similarity. When α is 0, the concept similarity is ignored and, at the other extreme, when α is 1, the keyword similarity is ignored.

$$\text{sim}(d, q) = ((1-\alpha) * \text{sim}(dk_j, qk) + (\alpha * \text{sim}(dc_j, qc))) \quad (4)$$

where

$$\text{sim}(dk_j, qk) = \frac{\vec{dk}_j \cdot \vec{qk}}{|\vec{dk}_j| \times |\vec{qk}|} \quad (5)$$

$$= \frac{\sum_{i=1}^t wt_{ij} \times wt_{iq}}{\sqrt{\sum_{i=1}^t wt_{ij}^2} \times \sqrt{\sum_{i=1}^t wt_{iq}^2}}$$

$$\text{sim}(dc_j, qc) = \frac{\vec{dc}_j \cdot \vec{qc}}{|\vec{dc}_j| \times |\vec{qc}|} \quad (6)$$

$$= \frac{\sum_{k=1}^c wt_{kj} \times wt_{kq}}{\sqrt{\sum_{k=1}^c wt_{kj}^2} \times \sqrt{\sum_{k=1}^c wt_{kq}^2}}$$

In the above equation, k represents the total number of keywords in the vocabulary and c represents the total number of concepts in the concept hierarchy.

Once the documents are scored, they are sorted and presented to the user in decreasing order by weight. In Section 4.2, we evaluate the effectiveness of conceptual search versus keyword-based search over a range of α values.

3.3 Conceptual Pruning

The conceptual information can also be used to remove documents that are completely unrelated to the user's concepts but which find their way into the top results due to a high keyword match. To do this pruning, the highest weighted concepts for each document in the result set are compared with the concepts provided by the user. The conceptual pruning algorithm exploits the hierarchical relationships between concepts to improve search results by removing documents from the result set that do not share the same parent or grandparent as the user-supplied concept(s). In Section 4.3, we evaluate the effectiveness of pruning as a post-process on search results retrieved using keyword-based search and conceptual search.

3.4 Prototype

Figures 2 and 3 show the web interface for the deployed version of KeyConcept. In Figure 2, the user's entry in the keyword section can be seen. In a normal search engine, this entry would be the only input the user would be able to provide. In KeyConcept, the user can also select the concept that he would like results to be most related to. Figure 3 shows the results for the search performed. The retrieval process involves a combination of keyword and conceptual search. The process may also include other factors for obtaining better results, the details of which are described in the following experiments.

4. EXPERIMENTAL EVALUATION

We design a series of experiments to evaluate the effectiveness of conceptual retrieval versus keyword-only based retrieval. We then evaluate the effectiveness of conceptual pruning used in combination with keyword search and conceptual search on a set of actual user queries.



KEYCONCEPT

A Conceptual Search Engine

DEMONS PEOPLE HOME API

Enter Keywords:

medical instruments

Enter the keywords you want to search for and select the categories you are looking for. You may select up to 10 categories.

Select Categories:

- Arts
- Business
- Computers
- Games
- Health
- Home
- News
- Recreation
- Reference
- Regional

Select Categories:

- Fitness
- Pharmacy
- Alternative
- Medicine
- Dentistry
- Nursing
- Nutrition
- Beauty
- Professions
- Occupational Health and Safety

Select Categories:

- Osteopathy
- Pharmacology

Select Categories:

- Directories
- Informatics
- Surgery

Fig. 2: KeyConcept Search Page



KEYCONCEPT

A Conceptual Search Engine

[DEMOS](#)
[PEOPLE](#)
[HOME](#)
[API](#)

Results :

Keywords Searched : medical instruments
Categories Selected : /HealthMedicine/Informatrics , /HealthMedicine/Surgery

Consumer Health Information
Weight : 0.770417 **Top 10 categories :** [View](#)
 The Cybirstspace Telemedical Office (sm) General Telemedical Services Medical Library Specialist: Resources Wellness Center Clinical Research Product Shopping Home HealthCare Nurse's Stat on Physician's Office Con...
[/id/Keyconcept/ire=AWT07.R17/93.html](#)

BAS Medical Unit
Weight : 0.759069 **Top 10 categories :** [View](#)
 BAS Medical Unit The BAS Medical Unit is managed by R&B Limited, a wholly owned subsidiary of the Robert Gordon University (RGU) in Aberdeen. The evolution of the unit which was formalised in 1989, paralleled...
[/id/Keyconcept/ire=AWT10.B14/61.html](#)

Internal Medicine
Weight : 0.755855 **Top 10 categories :** [View](#)
[Summary not available](#) ...
[/id/Keyconcept/ire=AWT19.B05/209.html](#)

About Dean Health System
Weight : 0.755698 **Top 10 categories :** [View](#)
 Dean Medical Center Dean Medical Center is the medical care component of Dean Health System "Dean Clinic" as it was originally called, has its roots in southern Wisconsin, serving patients since 1904. Dean Med...
[/id/Keyconcept/ire=AWT09.B37/295.html](#)

Internet Medical Resources
Weight : 0.75304 **Top 10 categories :** [View](#)
[Summary not available](#) ...

Fig. 3: KeyConcept Results Page

4.1 Experimental Setup

Because it is becoming a widely accepted shared resource, we chose to use the Open Directory Project hierarchy (ODP 2005) as our source for training data. As of April 2002, the Open Directory had more than 378,000 concepts. Because subtle differences between concepts may be apparent to a human but indistinguishable to a classification algorithm, we decided to use the concepts in the top three levels of the Open Directory only. For each concept, we downloaded all associated documents, up to a maximum of 50 per concept. This selection was done randomly. Thus, this initial training set contained 2,991 concepts and approximately 125,000 documents.

Before KeyConcept could be evaluated, several parameters had to be tuned to improve performance of the underlying classifier. No tuning was done using the test set documents or queries. For example, we had to determine the number of training documents needed for each concept. We also evaluated a variety of formulae for computing the concept term weights. Several experiments were performed using the using the ODP concepts and documents, and their details are reported in (Gauch et al. 2004b). To summarize the main points, we found that classification worked best when concepts were trained with between 30 and 40 documents. After this many documents are used, the accuracy plateaus, most likely because the most important vocabulary terms for the concept are already discovered. Based on the results, for these experiments, we used 30 training documents per concept and the formula shown in Section 3.1.

4.2 Evaluating Conceptual Retrieval

We designed two experiments to study whether a search engine that combines both 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 from the WT2g collection (Hawking et al. 1999). The test set included all of the documents judged relevant to any of the 50 test topics (2,279 documents) and 97,721 randomly selected documents to create a test corpus of 100,000 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 et al. 2001), we used the title section as the keyword query for both the keyword only and keyword plus concept searches. We tested two ways of associating concepts with the keyword queries generated from the titles. First, we

manually associated concepts with the queries based on reading the topic descriptions. Second, to demonstrate that this approach can also be used without burdening the users by requiring them to explicitly select the appropriate concept, we automatically associated concepts with the keyword queries. This automatic concept selection was performed by automatically classifying the description and narrative paragraphs included with each topic (average length 50 words) using our classifier, and using the top-weighted concepts thus identified. The relevance judgments supplied for each topic were used to evaluate the effectiveness of the system.

4.2.1 Measuring the relative importance of conceptual and keyword matches. In this experiment, we wanted to determine the relative importance to be given to conceptual versus keyword matches during retrieval. We varied two factors in this experiment: (1) the α factor, and (2) the number of concepts for each query that are actually used for document scoring. Because most users look only at the first page of search results, we evaluated performance by calculating the top-10 precision averaged across all queries. For all the experiments where significance values are reported, we performed a two tailed t-test with $\alpha = .05$ and the p-values are reported in the results of the experiment.

The baseline for this experiment is when $\alpha = 0$, i.e., only keyword matching is used. In this case, the precision obtained is 30.6%. As shown in Figure 4, the best result, 34.8%, is obtained when the top four concepts for each query are used and α is set to 0.4. This increase in precision from the baseline value was found to be significant ($p = .024$). Precision scores drop sharply for α values greater than 0.4, indicating that the conceptual score alone is not enough to achieve a good precision in searches. Precision also decreases when more than four concepts are used for retrieval. This reduction is likely because the additional concepts introduce noise. The evaluation was repeated using top 20 precision as the metric. In this case, the average precision reached a peak value of 26.9% for $\alpha = 0.3$, using the top three concepts for each query while it was 22% for $\alpha = 0$. Over a range of experiments, no significant difference were found between $\alpha = 0.3$ and $\alpha = 0.4$ or between 3 and 4 concepts per query. However, we observed that using $\alpha = 0.3$ with the top 3 concepts for each query gave the best results over the widest range of data. Thus, we decided to use these parameters as our baseline values for further experiments.

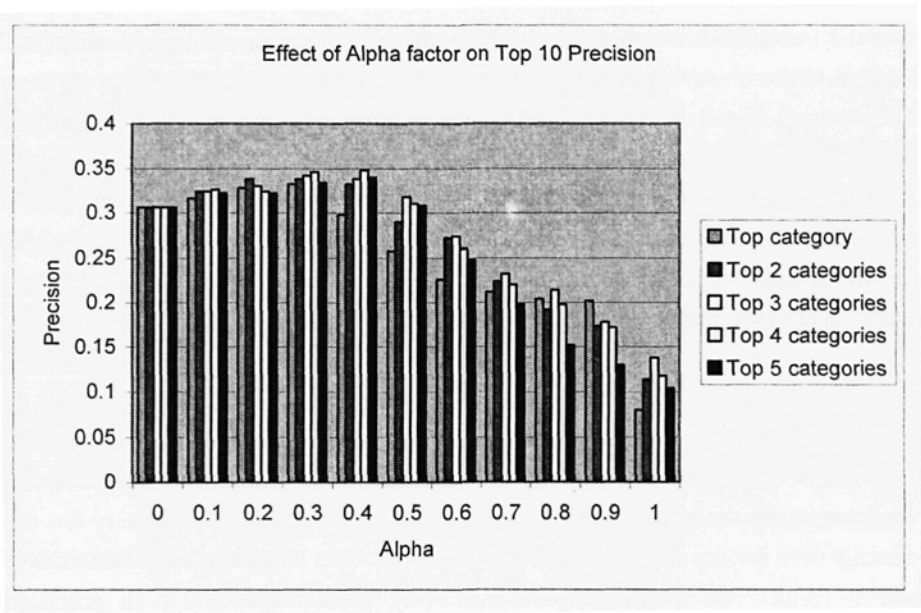


Fig. 4: Effect of α -Factor on Top 10 Precision

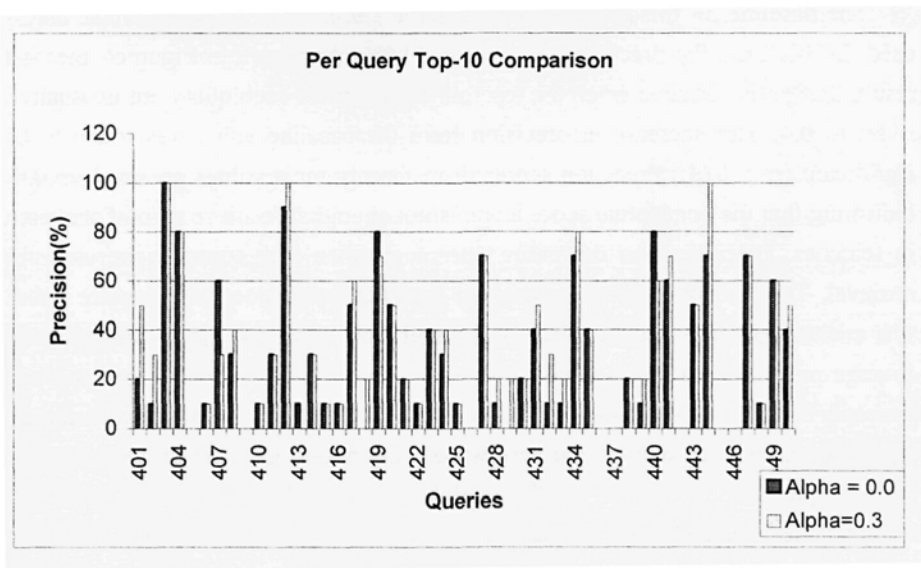


Fig. 5: Per-Query Top-10 Precision

4.2.2 Per query analysis of results. To better understand the effect of conceptual search, we examined the effect of conceptual retrieval on each of the 50 WT2g topics. We compared the results for the baseline, $\alpha = 0$ (keyword match only), and $\alpha = 0.3$ using top three concepts for each topic.

As shown in Figure 5, for the 50 queries available in the collection, 17 experienced an increase in precision when using combined keyword and concept retrieval, 3 experienced a decrease, and 30 showed no difference (including 9 for which no relevant results were retrieved by either version). We performed a two-tailed t-test on the results and found that, although small, the difference between keyword-only and keyword-concept retrieval performance is significant ($p\text{-value} = 0.018$). There were two queries (418 and 429) for which the use of combined keyword and conceptual retrieval retrieved results when none were returned from a pure word-based retrieval.

4.2.3 Discussion. We found that a value of 0.3 for the α factor increases the precision of the proposed searches when three concepts are used. This result indicates that the keyword matching is 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 because the correct concept may not always be the most similar, but is usually within the top 3 best-matching concepts returned by the classifier.

We believe the reason for the rather modest precision increases obtained in our experiments resides in the characteristics of the WT2g queries. For each of the test queries in this collection, human judges determined which documents were relevant for each query and their criteria were very restrictive. A document having both keyword and concept matches could be deemed non-relevant by some semantic or content detail obvious to a human evaluator, but which is undetectable for an automatic classifier. Many results that were on-topic, but did not contain the specific information desired as detailed in the narrative, were judged as non-relevant.

4.3 Evaluating Conceptual Pruning

We conducted experiments to evaluate the effects of exploiting the concept relationships to prune the result sets. For example, if a user entered a query “doors”

and supplied the concept “*music groups*”, the results would range from pages about rock music to pages about architectural design. We could eliminate results that were not about music by removing results that did not belong to the user-supplied concept. Two variations were performed: user-supplied concepts were compared with the highest weighted concepts for each document and matches were required to be found at the top level (Level 1) of the concept hierarchy; the experiments were rerun requiring matches at the second level, (Level 2). This approach enabled us to compare the effects of keeping results that generally matched the user’s concept (e.g., “Arts”) versus requiring a more specific match (e.g., “Arts/Music”). To check if pruning might be useful to obtain better search results, we compared these results to a simple keyword search and then with a conceptual search using $\alpha = 0.3$ as recommended by the experimental results shown in Section 4.2.

To evaluate the effectiveness of conceptual search for real-world use, we selected a set of test queries from the log file of a commercial search engine. Because conceptual search will likely benefit short, ambiguous queries more than longer, unambiguous queries, we randomly selected a total of 24 queries of different lengths: eight single-word queries; eight 2-word queries; and eight 3-word queries. Each query was associated with a single manually-assigned concept from the third level of the ODP tree. The relevance judgments, and related concepts, for these queries were created manually. The document collection used was the same WT2g collection used in the previous set of experiments. The top 10 documents were assessed for relevancy for each query.

4.3.1 Pruning the search results after simple keyword search. In this experiment, we manually selected a single concept for each of the 24 real-world test queries. For each query, a simple keyword search was performed on the query terms and any result obtained that did not belong to the same subtree in the concept hierarchy as the desired user concept was eliminated. Pruning was done at two different levels, the top level of the concept hierarchy and one level down. The results are shown in Table 1.

Based on a two-tailed *t*-test, the average precision increased significantly from 36.7% to 45.24% for level-1 pruning ($p = .00767$) and from 36.7% to 52.67% ($p = .0031$) for level-2 pruning. Although level-2 pruning outperformed level-1 pruning, the difference between them was not statistically significant. It also appears that for shorter 1 and 2 word queries, the level-2 pruning performs better. This finding makes intuitive sense because the shorter queries are likely to be more ambiguous and may require more aggressive filtering of the results.

TABLE 1

Precision comparison of keyword search vs. keyword search with pruning

Method	Queries (%)			
	1-word	2-word	3-word	Total
Keyword based retrieval	44	31	35	36.67
Keyword based retrieval with pruning at Level 1	48	38.35	49.38	45.24
Keyword based retrieval with pruning at Level 2	58	55	45	52.67

TABLE 2

Precision comparison of keyword + conceptual search vs.
keyword + conceptual with pruning

Method	Queries (%)			
	1-word	2-word	3-word	Total
Keyword + Conceptual Retrieval	54	49	51	51.33
Keyword + Conceptual Retrieval with Pruning at Level 1	63	58	62	61
Keyword + Conceptual Retrieval with Pruning at Level 2	71	65	65.26	67.08

TABLE 3

Precision Comparison of Retrieval and Pruning Methods

Method	Queries (%)			
	1-word	2-word	3-word	Total
Keyword Based Retrieval	44	31	35	36.67
Keyword Based Retrieval with Pruning at Level 2	58	55	45	52.67
Keyword + Conceptual Retrieval	54	49	51	51.33

4.3.2 Pruning the search results after conceptual search. This experiment compares the results of conceptual search with those obtained after pruning. The α value used for conceptual search was 0.3, as determined earlier. Table 2 shows the results obtained. There is a significant increase in precision for all three query sets.

The average precision increases from 51.33% to 61.0% for level-1 pruning ($p = 3.0 \times 10^{-4}$) and from 51.33% to 67.08% ($p = 1.35 \times 10^{-4}$) for level-2 pruning. In this case, level-2 pruning outperformed level-1 pruning for all three sets of queries and the increase in precision between level-1 and level-2 pruning was significant ($p = 7.32 \times 10^{-3}$).

4.3.3 Discussion. In the series of experiments described above, we see a dramatic increase in precision when pruning is used. Table 3 illustrates the effectiveness of the different methods, as measured by top-10 precision. As expected, a combination of keyword retrieval, conceptual retrieval, and pruning based on related concepts, obtains the best results. The average precision for keyword and conceptual retrieval with level-2 pruning, over all 24 queries, is 67.1% versus a baseline of only 36.7% for keyword based retrieval.

The average total precision is above 60% in all three sets of queries tested, i.e., single-word, 2-word, and 3-word queries. Counter-intuitively, single-word queries were more precise than longer queries. This occurs perhaps because the additional words in the longer queries are adding noise rather than increasing search result accuracy, i.e., the results sets include documents highly relevant to the individual words in the longer queries rather than selecting documents relevant to all query words. It also appears that using conceptual information during retrieval and for pruning is complementary. Comparable increases in precision are observed when using pruning alone and when performing conceptual search. When pruning is combined with conceptual search, however, the effect appears to be additive. This observation confirms our belief that pruning is an important way to eliminate off-topic results, particularly when combined with conceptual retrieval to identify a larger pool of on-topic results.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented KeyConcept, a search engine that indexes documents with their concepts from the Open Directory Project in addition to their keywords. When querying the search engine, the conceptual information is used, in addition to the keywords, to provide results that are more relevant to the user. The paper also examined methods to exploit the hierarchical relationships between concepts to further improve search results via pruning.

In the conceptual retrieval experiments with the WT2g topics, we obtained a top-10 precision of 34.8% versus 30.6% for keyword search alone, a significant improvement. With the addition of pruning on a set of 24 actual queries submitted to a search engine by end-users, of queries, the best result of 71% top-10 precision, occurred on our single-word query set when conceptual retrieval was performed with $\alpha = 0.3$ and results when pruned at level 2. The top-10 precision for our entire 24-query set increases from 36.7% for traditional keyword based search to 63.77% for conceptual search ($\alpha = 0.3$) with level 2 pruning, a significant improvement ($p = .0038$). Based on the results, we conclude that concept matches can be used to significantly increase search accuracy over keyword search alone. Significantly improved results are observed when concept matches are used to modify document weights (conceptual search) and when they are used to prune off-topic results. When the two approaches are used together (conceptual search with pruning), an additional significant gain in search accuracy is obtained, so the two approaches appear to be complementary.

Currently, the concepts for each query must be provided manually, or by classifying text associated with the query. Because it has been shown that users find it difficult to provide additional information (Wærn, 2004), we plan to combine the KeyConcept results with our work on user profiling (Trajkova & Gauch 2004) so that the concepts are supplied implicitly via user profiles. If successful, KeyConcept could be an effective, personalized search engine. We have also conducted experiments on a more accurate hierarchical classifier (Pulijala & Gauch 2004) that we plan to incorporate. The more accurate the document classification process is, the more accurate the conceptual matching will be. Another area to investigate would be a comparison of conceptual indexing based on ontologies versus those based on query expansion. Finally, for this approach to the work, there must be a good match between the concept hierarchy used and the document collection. We are investigating ways to identify good document collection-concept hierarchy matches, and also to adapt concept hierarchies to match the underlying collection better.

6. REFERENCES

- Aggarwal, C. and Yu, P. 2001. On Effective Conceptual Indexing and Similarity Search in Text Data, in: Proceedings of IEEE International Conference on Data Mining (ICDM Conference), San Jose, California, USA, November/December 2001, 3-10.

- Allen, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft R.W. et al. 2003. Challenges in Information Retrieval and Language Modeling, *ACM SIGIR Forum, Spring 2003*, 37(1), 31-47.
- Baziz, M., Boughanem, M., and Aussenac-Gilles, N. 2005. Semantic networks for a conceptual indexing of documents in IR, *Proceedings of the 7th ISPS' Algiers*, May 2005.
- Cesarano, C., d'Acierno, A., and Picariello, A. 2003. An intelligent search agent system for semantic information retrieval on the internet, *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*, New Orleans, USA, 2003, 111-7.
- Chaffee, J. and Gauch, S. 2000. Personal ontologies for web navigation, *Proceedings of the 9th International Conference on Information Knowledge Management (CIKM)*, Mclean, Virginia, USA, November 2000, 227-234.
- Chang, Y., Kim, M. and Raghavan, V. 2006. Construction of query concepts based on feature clustering of documents, *Information Retrieval*, 9(3), June 2006.
- Chekuri, C., Goldwasser, M., Raghavan, P., and Upfal, E. 1997. Web Search using automatic classification, *Proceedings of the 6th International WWW Conference*, Santa Clara (CA), USA, April 1997.
- Dasarathy, B. V. 1991. *Nearest Neighbor (NN) norms: NN Pattern Classification Techniques*. McGraw-Hill Computer Science Series. IEEE Computer Society Press, Las Alamitos, California.
- Dhillon, I.S., Mallela, S., and Kumar, R. 2002. Enhanced word clustering for hierarchical text classification, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (March), 191-200.
- Dumais, S., and Chen, H. 2000. Hierarchical classification of Web content, *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, Athens, Greece, 256-63.
- Ferguson, T. S. 1973. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 209-30.
- Gauch, S., Wang, J., and Rachakonda, S. 1999. A corpus analysis approach for automatic query expansion and its extension to multiple databases. *ACM Transactions on Information Systems*, July 1999, 17(3), 250-69.
- Gauch, S., Chaffee, J., and Pretschner, A. 2004a. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, 1(3-4), 219-34.
- Gauch, S., Ravindran, D., Induri, S., Madrid, J., and Chadalavada, S. 2004b. *KeyConcept – A conceptual search engine*. Internal Technical Report ITTC-FY2004-TR-8646-37, Information and Telecommunication Technology Center, University of Kansas. Available at: <http://www.ittc.ku.edu/keyconcept/publications/ITTC2003.pdf> (accessed 10 July 2005).
- Gauch, S., Chandramouli, A., and Ranganathan, S. 2006. *Training a hierarchical classifier using inter-document relationships*. Technical Report ITTC-FY2006-TR-31020-01, Information and Telecommunication Technology Center, University of Kansas.

- Guarino, N., Masolo, C., and Vetere, G. 1999. OntoSeek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3), 70-80.
- Gulli, A. and Signorini, A. 2005. The indexable web is more than 11.5 billion pages, *International World Wide Web Conference*, Chiba, Japan, 2005, 902-3.
- Hawking, D., Voorhees, E., Craswell, N., and Bailey, P. 1999. Overview of the TREC8 Web Track, *Eighth Text Retrieval Conference (TREC-7)*, Maryland, USA. Available at: <http://citeseer.ist.psu.edu/hawking00overview.html> (accessed 10 July 2005).
- Heflin, J. and Hendler, J. 2000. Dynamic ontologies on the web, *Proceedings of 17th National Conference on Artificial Intelligence(AAI)*, Menlo Park, California, USA, 2000, 443-9.
- Jansen, B. and Spink, A. 2006. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. *Information Processing & Management*, 42(1), 248-63
- Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features, *Proceedings of the European Conference on Machine Learning*, Springer.
- Klink, S., Hust, A., Junker, M., and Dengel, A. 2002. Collaborative learning of term-based concepts for automatic query expansion, *Proceedings of ECML 2002, 13th European Conference on Machine Learning*, Helsinki, Finland, August, 2002, 195-206.
- Knight, K. and Luk, S.K. 1994. Building a large-scale knowledge base for machine translation, *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, Seattle, Washington, USA, July/August 1994, 1, 773-8.
- Krovetz, R., and Croft, B. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2), 115-41.
- Labrou, Y. and Finin, T. 1999. Yahoo! As an ontology – Using Yahoo! Categories to describe documents, *Proceedings of the 8th International Conference On Information Knowledge Management (CIKM)*, Kansas City, USA, 180-187.
- Liu, F., Yu, C., and Meng, W. 2002. Personalized web search by mapping user queries to categories, *Proceedings of the 11th International Conference on Information and Knowledge management*, McLean, Virginia, USA, 55-65.
- ODP 2005 Open Directory Project. Available at: <http://dmoz.org> (accessed July 10th 2005).
- Pearce, C. and Miller, E. 1997. The TellTale dynamic hypertext environment: Approaches to scalability. *Advances in Intelligent Hypertext, Lecture Notes in Computer Science*. Springer-Verlag, 1997.
- Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., and Breuel, T. 2002. The consumer side of search: Personalized search. *Communications of the ACM*, 50-55.
- Pretschner, A. and Gauch, S. 1999. Ontology Based Personalized Search, *Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'99)*, Chicago, IL, USA, 391-8.

- Pulijala, A. and Gauch, S. 2004. Hierarchical text classification, *International Conference on Cybernetics and Information Technologies, CITSA 2004*, Orlando, Florida, USA.
- Rocchio, J. 1971. Relevant feedback in information retrieval, in: *The smart retrieval system - experiments in automatic document processing*, edited by Salton, G., Englewood Cliffs, NJ.
- Sieg, A., Mobasher, B., Lytinen, S., and Burke, R. 2003. Concept based query enhancement in ARCH, *Proceedings of the International Conference on Internet Computing 2003 (IC'2003)*, Las Vegas, Nevada, USA, 613-9.
- Trajkova, J., and Gauch, S. 2004. Improving ontology-based user profiles, *RIAO 2004*, Vaucluse, France, 380-9.
- Tirri, H. 2003. Search in vain: challenges for Internet search. *Computer*, **36**(1), 115-6.
- Vapnik, V. 2000. *The Nature of Statistical Learning Theory*, 2nd Edition, Springer, New York.
- Yahoo. 2005. Yahoo! <http://www.yahoo.com> (accessed July 10th, 2006).
- Yahoo. 2006. Yahoo! Mindset <http://mindset.research.yahoo.com/> (accessed July 10th, 2006).
- Zhang, W., Qian, H., Mao, W. and Sun, G. 1999. A multilingual (Chinese, English) indexing, retrieval, searching search engine, *Proceedings of the 9th INET Conference*, San Jose, CA, USA.
- Zhu, X., Gauch, S., Gerhard, L., Kral, N. and Pretschner, A. 1999. Ontology-based web site mapping for information exploration, *Proceedings of the 8th International Conference On Information Knowledge Management (CIKM)*, Kansas City, USA, 188-94.
- Zien, J., Meyer, J., Tomlin, J. and Liu, J. 2001. Web query characteristics and their implications on search engines, *Proceedings of the 10th International WWW Conference*. Hong Kong, China.