

Evaluating Ontology Based Search Strategies

Chris Loer¹, Harman Singh², Allen Cheung³, Sergio Guadarrama⁴, and Masoud Nikravesh⁵

¹ email: cloer@cal.berkeley.edu

² email: hjsingh@berkeley.edu

³ email: allenmhc@cal.berkeley.edu

⁴ Dept. of Artificial Intelligence and Computer Science,
Universidad Politécnica de Madrid
Madrid, Spain

email: sguada@dia.fi.upm.es

⁵ Berkeley Initiative in Soft Computing (BISC)

Computer Science Division, Dept. of EECS

University of California

Berkeley CA 94704, USA

email: nikravesh@cs.berkeley.edu

Abstract. We present a framework system for evaluating the effectiveness of various types of “ontologies” to improve information retrieval. We use the system to demonstrate the effectiveness of simple natural language-based ontologies in improving search results and have made provisions for using this framework to test more advanced ontological systems, with the eventual goal of implementing these systems to produce better search results, either in restricted search domains or in a more generalized domain such as the World Wide Web.

1 Introduction and Motivation

Soft computing ideas have many possible applications to document retrieval and internet search, but the complexity of search tools, as well as the prohibitive size of the Internet (for which improved search technology is especially important), makes it difficult to test the effectiveness of soft computing ideas quickly (specifically, the usage of conceptual fuzzy set ontologies) to improve search results. To lessen the difficulty and tediousness of testing these ideas, we have developed a framework for testing the application of soft computing ideas to the problem of information retrieval ¹. Our framework system is loosely based on the “General Text Parser (GTP)” developed at the University of Tennessee and guided by “Understanding Search Engines”, written by some of the authors of GTP [1]. This framework allows a user to take a set of documents, form a “vector search space” out of these documents, and then run queries within that search space.

¹ This project was developed under the auspices of the Berkeley Initiative in Soft Computing from January to May of 2004

Throughout this paper, we will use the term “ontology” to refer to a data structure that encodes the relationships between a set of terms and concepts.

Our framework takes an ontology and uses its set of relationships to modify the term-frequency values of every document² in its search space, with the goal of creating a search space where documents are grouped by semantic similarity rather than by simple coincidence of terms. The interface for specifying an ontology to this framework is a “Conceptual Fuzzy Set Network,” which is essentially a graph with terms and concepts as nodes and relations (which include activation functions) as the edges respectively[8].

As well as allowing users to directly manipulate a number of factors that control how the system indexes documents (i.e. the ontology that the system uses), the framework is specifically designed to be easily extensible and modular. We believe that there are a wealth of strategies for improving search results that have yet to be tested, and hope that for many of them, simple modifications to this framework will allow researchers to quickly evaluate the utility of the strategy.

2 System Description

The framework exists as a set of packages for dealing with various search tasks: it is currently tied together by a user interface that coordinates the packages into a simple search tool. This section will give a brief overview of the interesting features of the framework – for more detailed documentation of both the features and the underlying code, please see <http://www-bisc.cs.berkeley.edu/ontologysearch>. While making design decisions, we have tried to make every part of the code as extensible and modular as possible, so that further modifications to individual parts of the document indexing process can be made as easily as possible, usually without modifying the existing code save a few additions to the user interface. Our current implementation includes the following features:

- A web page parser with a word stemmer attached
- Latent Semantic Indexing (LSI) of a vector search space
- Linear “Fuzzification” based on an ontology specified in XML
- The ability to run queries on a defined search space created from a set of documents
- A visualization tool that projects an n-dimensional search space into two dimensions
- Fuzzy c-means clustering of documents
- Automatic generation of ontologies based on OMCSNet

² That is, the number of times a given term appears in a given document, for all terms. The vector of terms for every document is normalized for ease of calculation.

2.1 Search Spaces

After parsing, each document is represented as a vector mapping terms to frequencies, where the frequency “value” is measured with Term-Frequency Inverse Document Frequency (TF-IDF) indexing (although the system allows for alternative frequency measurements). These documents are represented as an n -dimensional vector space, where n is the number of unique terms in all of the documents, i.e. the union of all terms in all documents. From this initial vector space it is possible to construct an “LSI Space”, which is a copy of the original vector space that has been modified using LSI; in our case, we use a Singular Value Decomposition (SVD) matrix decomposition method to optimally compress sparse matrices³. SVD compression is lossy, but the optimality of the compression ensures that semantically similar terms are the first to be conflated as the amount of compression increases [1]. Query matching is performed by calculating the cosine similarity between the query term vector and document term vectors within the search space.

2.2 Ontology Implementation

Our framework treats “ontologies” as a completely separate module, and its only requirement is that an ontology must be able to “fuzzify” a set of terms (i.e. relate terms to each other) according to its own rules. We have included an ontology parser which parses XML files of a certain format into a base ontology class. Figure 1 shows an example of the XML format of a simple ontology; this basic ontology class stores a set of words and for each word, a set of directed relations from that word to all other words in the ontology⁴.

To reshape a search space using an ontology, the user must choose an activation function for increasing or decreasing the value of related terms as specified by the rules of the ontology. With our framework, we have included a linear propagation function for ontologies; it takes the frequency of every term in the document, looks for that term in the ontology, and increases the frequency of all related terms by the value specified in that ontology⁵. If sigmoid propagation is being used, then frequencies will actually be decreased if they fall below a certain threshold, so that only terms that have a high degree of “support” (that is, they occur along with other terms that are deemed to be related to them, and thus probably have to do with the central meaning of the document) end up becoming amplified.

³ Our vector space is a sparse matrix, as every document has only a fraction of all the terms in the search domain

⁴ Each relation contains a real value between 0 and 1, where 0 signifies a complete lack of relation and 1 signifies synonyms.

⁵ For example, given that **farm** is related to **agriculture** by 0.45, **farm** has a frequency value of 2, and **agriculture** has a value of 0, linear propagation would give **agriculture** a new value of $0.45 * 2 = 0.9$.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE Ontology (View Source for full doctype...)>
- <Topic>
- <term word="right">
  <relation word="right wrong" weight="0.5" />
</term>
- <term word="bush">
  <relation word="on table" weight="0.4375" />
  <relation word="live underwater" weight="0.4375" />
  <relation word="tree" weight="0.7890625" />
  <relation word="compute" weight="0.25" />
  <relation word="at grocery store" weight="0.25" />
  <relation word="in horse mouth" weight="0.25" />
  <relation word="be yellow green and red" weight="0.25" />
  <relation word="stem" weight="0.68359375" />
  <relation word="hide in" weight="0.5" />
  <relation word="patriotism" weight="0.4375" />

```

Fig. 1. A simple example of the format of an ontology stored in XML. This particular ontology is automatically generated from a database of concepts, and thus has many relations that do not immediately seem useful.

2.3 Clustering

The framework includes a clustering unit that performs Fuzzy C-Means clustering on a search space[2]; the user interface allows users to specify whether to perform clustering, how many clusters to create, and what membership threshold to use. The clustering process assigns each document a degree of membership in each of the clusters, used in the visualization to illustrate document groupings (which should tend to correspond with the groupings that can be visually perceived in the two dimensional representation) as well as in query execution to speed up processing queries: with clusters, the system trims the document space by looking only at documents that have a relatively high degree of membership in the cluster that best fits the query.

2.4 Visualization

The user interface has a visualization tool which plots a two dimensional representation of the documents in the search space. LSI is used to obtain a rank-2 decomposition of the n -dimensional search space, which is ultimately a $2 \times n$ matrix of points in two dimensions. Our interface plots that matrix, allows the user to move around and inspect documents, and colorizes documents by their degree of membership in any given cluster. The aim of the tool is to allow users to quickly determine the salient characteristics of a search space and to determine, on a very broad level, how the use of an ontology affects the space. The user may also plot a two dimensional representation

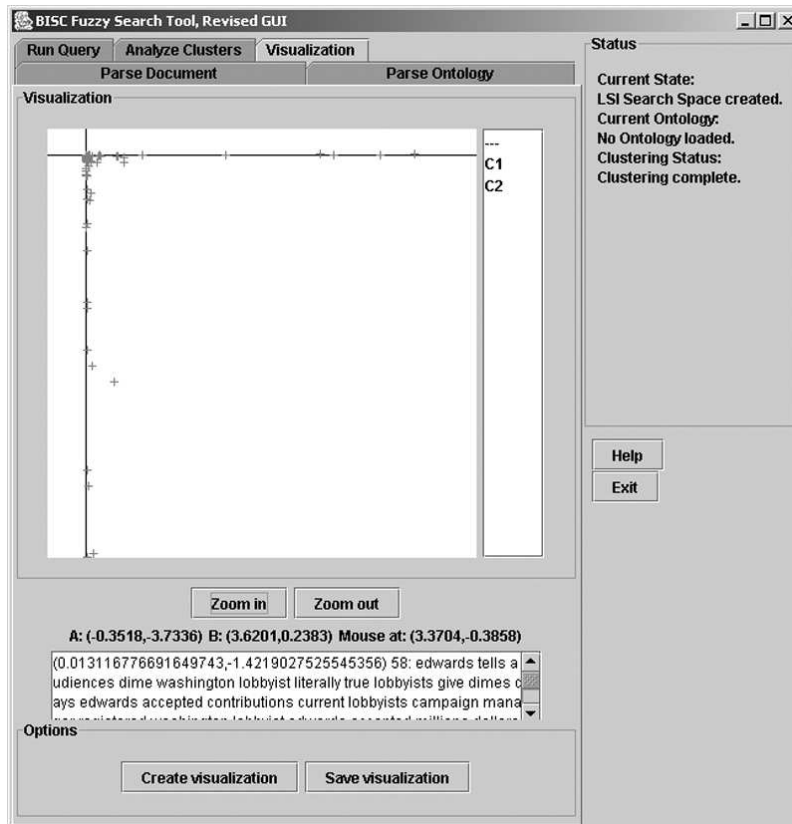


Fig. 2. Two dimensional visualization of a search space containing documents in Spanish and English. English documents cluster on the y axis, while Spanish documents cluster on the x axis.

of terms (which is based on the same underlying search space), in order to determine how certain terms might group together.

2.5 Ontology Generation

A simple tool has been built into the system that takes a search space, finds the most common terms in that search space, and then constructs an ontology out of information available through MIT's Open Mind Common Sense corpus (OMCSNet) found at <http://web.media.mit.edu/~hugo/conceptnet/>. This ontology simply encodes all of the relations to various words and phrases that OMCSNet has for the common terms. Admittedly, this tool is crude, in that it contains no domain specific information and includes irrelevant phrases which are meaningless when broken down into individual terms, it provides a good initial approximation of an ontology for testing.

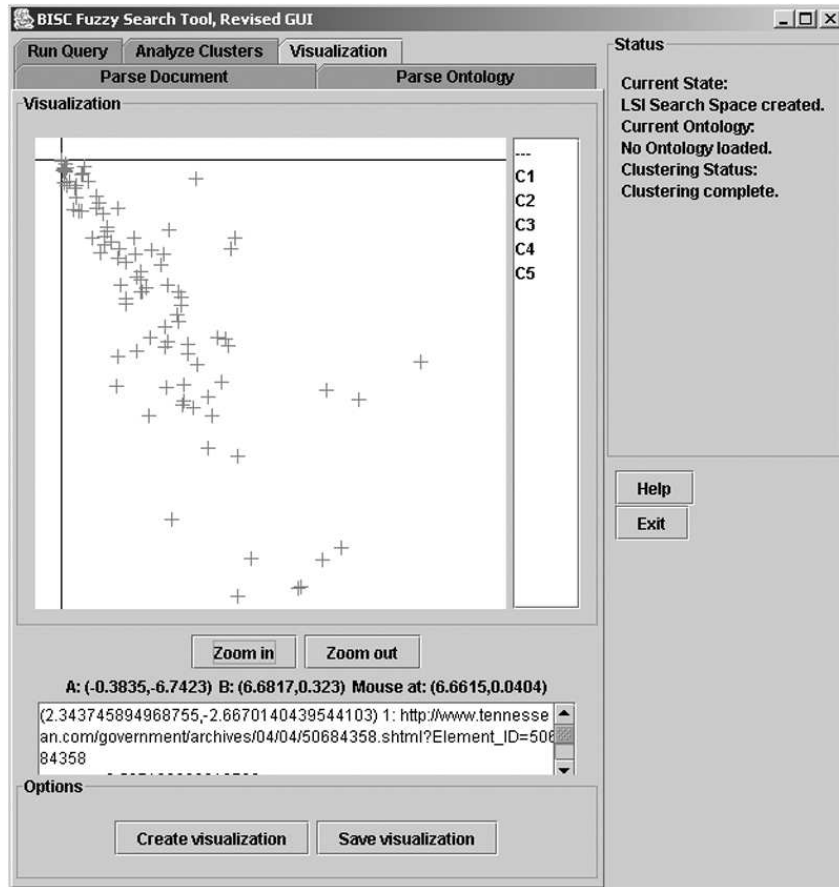


Fig. 3. Two-dimensional visualization of a search space containing documents related to the war in Iraq. No ontological modifications have been made to the search space.

3 Results

To test the operation of our system, we created search spaces with sets of documents drawn from Internet news sites. As an initial test of our two dimensional visualization, we first sampled a number of news sites in two distinct languages: Spanish and English. As we expected, the visualization displayed two completely distinct clusters of documents, as shown in Figure 2. We ran our next set of tests on a body of 124 news articles retrieved from GoogleTMNews by searching for the terms “Iraq War”. To generate an ontology for testing, we found the most common terms in a set of documents and went to OMCSNet to create an ontology (as mentioned, this functionality is already built into our framework). The ontology we generate thus has

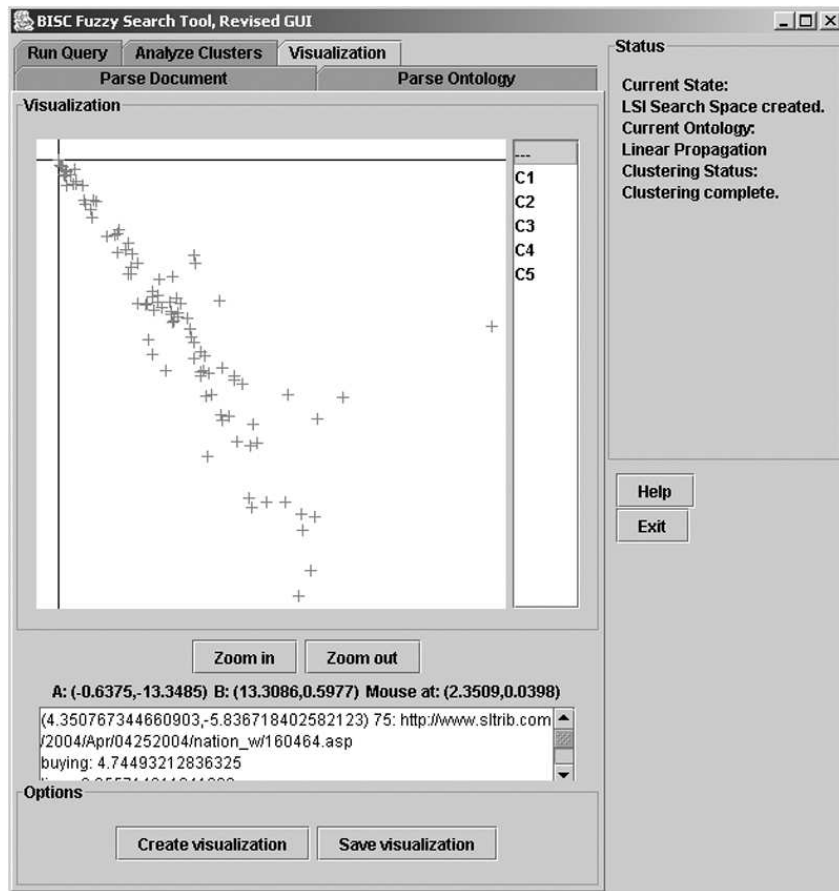


Fig. 4. Two-dimensional visualization of the same search space after “fuzzification” with an English-language ontology based on the most common words in the document set.

no domain-specific information and no notion of abstract concepts; it only encodes the relationship between (hopefully relevant) English terms and other terms that may show up in our search space. Despite this simplicity, our expectation was that adjusting term frequency values would make documents which referred to similar topics appear more similar.

3.1 Visualization

We used our visualization tool to get a two dimensional representation of the search space before and after modifying term frequencies with our ontology. Figure 3 shows the visualization before modification, and Figure 4 shows the visualization after modification.

Our OMCSNet ontology only increased term frequencies, and did so regardless of context such that the degree of similarity would only increase between any two documents after modification. Not surprisingly, the visualization after modification showed that all documents were more tightly clustered. We hypothesize that, even though all documents become more similar to each other, topic-related documents see a *greater* effect from the ontological transformation and pull even closer together. Although we were able to inspect visual points to informally verify that similar documents were in fact near each other, we had no systematic way to evaluate the effectiveness of the transformation. If we were to use an advanced ontology which took note of context, we would expect to see a greater impact in the visualization. In general, because of the coarse nature of visuals and the high level of rank reduction we need to obtain a two-dimensional representation, our visualization tool only provides an intuition for how the documents are related and the overall effect of an ontology, but cannot give any systematic evidence for the effectiveness of a given ontology.

3.2 Queries

To measure the effectiveness of an ontology at improving information retrieval for a body of documents, we compared search results from a variety of queries on a given corpora with and without the use of an ontology. The accuracy of our results are subjective; having no objective standard to measure our results against, we cannot give concise numbers on how well our search framework performs short of developing a point-based rubric to manually evaluate, rank and compare search results.

As our primary interest while writing this system was verifying that the system itself performed as expected, we did not develop any ranking system for accuracy, but rather evaluated results of several test queries based on informal observation, comparing the use of different ontologies (including the “null” ontology). Figures 5 and 6 show the results of searching for the term “patriotism” before and after ontologically modifying an “Iraq War” search space.

The top results shown in Figure 5 are documents containing the term “patriotism”, and they are separated from lower results by a steep drop in similarity index value. Upon further inspection, our lower-ranked documents do not include the term “patriotism” but still seem to hold some relation to the term: this may be an effect of semantic information captured by LSI, or it may simply be that all documents related to the Iraq War are related to patriotism in some way.

In Figure 6, the first document to contain the word “patriotism” is actually ranked in the middle of the list of query results, but we see that higher-ranking documents do discuss the *display* of patriotism, including phrases such as “flag waving”, “supporting veterans”, and “national pride”. It is of course easy to make up a story to explain why the results in one figure are

Index	Vector as String
0.04142...	1: http://www.tennessean.com/government/archives/04/04/...
0.03148...	103: http://www.opendemocracy.net/debates/article-2-103-1...
0.02209...	84: http://www.euobserver.com/index.phtml?aid=15359plan...
0.01925...	65: http://www.menafn.com/qn_news_story_s.asp?StoryId=...
0.01589...	16: http://www.zwire.com/site/news.cfm?BRD=1078&dept_i...
0.01219...	62: http://rockland.villagesoup.com/Community/Story.cfm?S...
0.01056...	95: http://straittimes.asia1.com.sg/news/story/0,4386,246...
0.00998...	31: http://timesargus.com/Local/Story/83008.htmlsinging: 4...
0.00863...	9: http://www.tomahjournal.com/articles/2004/05/02/opinion...
0.00819...	69: http://www.abc.net.au/lateline/content/2004/s1096064.ht...
0.00805...	71: http://breakingnews.iol.ie/news/story.asp?j=102330464...
0.00747...	58: http://www.infoshop.org/inews/stories.php?story=04/04/...
0.00658...	59: http://www.omaha.com/index.php?u_np=0&u_pg=54&u...
0.00648...	12: http://www.charleston.net/stories/050204/ter_02bushra...
0.00606...	38: http://www.rockymountainnews.com/drmn/opinion/articl...
0.00531...	0: http://www.dissidentvoice.org/May2004/Petersen0501.ht...
0.00517...	8: http://www.theaustralian.news.com.au/common/story_pa...
0.00488...	90: http://www.zwire.com/site/news.cfm?BRD=1699&dept_i...
0.00428...	89: http://www.in-forum.com/articles/index.cfm?id=56645&s...

Fig. 5. The result of a search for “patriotism” before modifying the search space.

better than the results of the other: this data is not meant to be evidence that the ontology we used actually improved search results, but rather to demonstrate how an ontology changes the results of our query and how the system allows the user to quickly compare search spaces with and without the help of ontologies.

4 Future Work

We have identified several projects that could be pursued using the framework, either as extensions or as tests performed within the system. Our desire to test some of these ideas motivated the design of this system, but we expect that the framework may prove useful for testing ideas that never occurred to us.

4.1 Hierarchical Conceptual Fuzzy Sets

The framework lends itself to a more advanced notion of “ontology” than we have used in our initial implementation, which focuses simply on direct

Index	Vector as String
0.06626...	16: http://www.zwire.com/site/news.cfm?BRD=1078&dept_i...
0.06237...	106: http://www.myrtlebeachonline.com/mld/myrtlebeachonli...
0.06032...	44: http://english.peopledaily.com.cn/200404/30/eng200404...
0.05887...	27: http://www.capitolhillblue.com/artman/publish/article_44...
0.05829...	37: http://www.voanews.com/article.cfm?objectID=088BEAF...
0.05810...	62: http://rockland.villagesoup.com/Community/Story.cfm?St...
0.05751...	66: http://www.abc.net.au/ra/newstories/RANewsStories_10...
0.05735...	38: http://www.rockymountainnews.com/drmn/opinion/article...
0.05710...	65: http://www.menafn.com/qn_news_story_s.asp?StoryId=...
0.05587...	42: http://news.xinhuanet.com/english/2004-04/29/content_1...
0.05443...	29: http://www.abs-cbnnews.com/NewsStory.aspx?section=...
0.05352...	9: http://www.tomahjournal.com/articles/2004/05/02/opinion/...
0.05302...	1: http://www.tennessean.com/government/archives/04/04/5...
0.05286...	71: http://breakingnews.iol.ie/news/story.asp?j=102330464...
0.05239...	36: http://www.theaustralian.news.com.au/common/story_p...
0.05236...	80: http://www.sunherald.com/mld/sunherald/living/8514353...
0.05091...	79: http://washingtontimes.com/upi-breaking/20040426-024...
0.05088...	28: http://www.scoop.co.nz/mason/stories/P00405/S00007...
0.05082...	21: http://www.sunherald.com/mld/sunherald/living/8570...

Fig. 6. The result of the same search after modification. Note that all results have received a higher similarity index as a result of “fuzzification” and the order of relevance has changed from the query without an ontology.

relationships between terms. To capture semantic information with greater depth, hierarchical conceptual fuzzy sets may prove useful. A hierarchical conceptual fuzzy set network specifies concepts and relations using multiple levels of abstraction. For example, the term **Porsche** would trigger activation of the concept **Sports Cars**, which would in turn activate **Cars**, and then **Moving Vehicles**. In this situation the additional term **Ferrari** would strongly trigger **Sports Car**, while the term **Truck** would strongly trigger the broader **Moving Vehicles**. By more accurately determining the context of words within a document, and thus the “meaning” of the document, the use of a hierarchical conceptual fuzzy set network could further improve the quality of query results.

Extending the framework to test this idea would require an extension to the current ontology parser class, an extension to the current ontology class, and a method to create appropriate “hierarchical ontologies” for testing purposes.

4.2 Tailored Ontologies

Because this tool allows us to compare query results with and without ontological information, it will be useful for testing the effectiveness of various ontologies at capturing semantic structure within a search domain. Possible experiments include:

- Hand craft a set of relations among words related to an academic discipline (i.e. Computer Science), and then use the ontology to search in the domain of technical articles for that discipline.
- Automatically generate an ontology for a search domain based a set of criteria (i.e. term coincidence) and compare results with and without this ontology. Specifically, it may be interesting to evaluate the effectiveness of “fuzzy thesauri” [4] generated from the World Wide Web.
- Test the utility of feedback-driven ontology systems (i.e. the BISC Image Search program). User feedback could be used to create an interactive system that personalizes context in queries ⁶ or to create a term-centered (rather than phrase-centered) general knowledge base on commonly assumed word relations.
- Automatically create ontologies based on semantic information from a natural language database. ⁷ While we have already implemented a tool to generate ontologies from MIT’s OMCSNet, a context-sensitive ontology from a language database has further applications. That is, with the capacity to parse sentences, it is possible determine which terms and concepts in a document should be activated with a higher degree of accuracy. In the other direction, document-wide term-frequencies along with the activation values of terms and concepts in a conceptual fuzzy set network can aid a natural language parser in resolving ambiguous contexts.

As mentioned in Section 3.2, analysis and evaluation would require a standardized rubric for ranking the quality of results.

4.3 Alternative Frequency Measures

Throughout our program, we have used Term Frequency-Inverse Document Frequency (TF-IDF) measures, but we have not experimented with Non-monotonic Document Frequency (NMDF) measures [5], term ranking algorithms based on evolutionary computing, or other methods for measuring the occurrence of terms within documents. Unfortunately, because indexing methods often rely on detailed information from document parsing, modifications and additions to the modules that deal with indexing will most likely

⁶ For example, a system would detect that its user uses the terms “Bush” and “president” interchangeably and tighten the ontological relationship between these two terms.

⁷ Examples include Princeton’s WordNet, MIT’s OMCSNet, and Berkeley’s FrameNet.

require changes in the parsing modules as well, violating the abstraction barriers and modularity of the framework.

4.4 Integration with GoogleTMSearch

We currently parse a manually entered set of web pages or text documents. If we take this idea one step further and try to include some form of automated document search and extraction on the World Wide Web, a natural direction is making use of GoogleTM's search engine (via their free API) to download new documents on-the-fly. A typical scenario would have a user searching for documents about "car repair"; the system would fetch a group of documents related to automotive maintenance by using GoogleTM's search API, parse them as a document search space, and query within this tight domain of documents (perhaps also using a tailored ontology using automated methods such as Section 4.2's OMCSNet-ontology creation).

4.5 Query Refinement and Expansion

With the World Wide Web (accessed via Section 4.4's methods or some other means) at our fingertips, we should be able to make refinements of queries or expand queries to include other relevant documents and enlarge the size of our search space. For instance, to follow-up and expand on Section 4.4 and integration with GoogleTM, we can use the following algorithm:

```

Input query
Use GoogleTMAPI to find top  $n$  results for the terms in the query
Use OMCSNet to create context-specific ontology based on the most
common terms in the top  $n$  results
Use OMCSNet ontology to find the top  $j$  groups of terms most related
to the terms of the query
Use GoogleTMAPI to find top  $n$  results for each of the groups of terms
related to the query
Add all documents retrieved from GoogleTMto search space
Reorder documents from API search using OMCSNet ontology
Return top  $k$  most highly ranked documents

```

Such a process would in effect expand and refine our search space – by sampling multiple parts of the World Wide Web with the help of an ontology, we are expanding beyond the limited number of terms in the user's query, and by reordering documents with respect to the ontology, we are refining our results and giving higher ranks to documents which are closely related to the query. Abstractly, we are approximating "fuzzification" of our perspective of the World Wide Web with the ontology. If we had re-indexed the entire World Wide Web, documents using ontologically related terms would

be similar to each other: although these documents might have no relation to each other that would show up in a GoogleTM search, this process would ensure that all relevant documents would be fetched and the reordered such that the results would be similar to the results we would expect if we had completely re-indexed.

4.6 Commercial Applications

We designed the system to work well on relatively small corpora, in the range of thousands of documents. If tailored and domain-specific ontologies prove to be a useful technique in improving search results within a restricted domain, these ideas could be applied to a scalable search system⁸ based on some of the principles laid out by Brin and Page Brin:1997. Such a system would still probably be best suited to searching restricted (and more structured) domains, rather than the entirety of the World Wide Web, and might be used by libraries and other institutions charged with storing academic or professional knowledge to provide improved search results to their clients.

5 Conclusion

As this ontology based search system is primarily a tool for testing new ideas, this paper's intention is to create awareness of the availability of this tool. For the interested reader, the complete source code for this system, as well as documentation, is available at <http://www-bisc.cs.berkeley.edu/ontologysearch>. Although completely different indexing techniques would be necessary to efficiently apply ontology-based ideas to the task of searching very large corpora, we believe that this system could serve as a fair prototype for a tool to search specific, limited-size corpora (i.e. a set of books or papers in a particular field). If we are able to find and develop a method of capturing semantic significance of documents at this level, this technology then could be expanded into the larger domain of generalized Internet search.

Acknowledgements

The ideas in this project came from the lecture series presented by Dr. Masoud Nikravesh to the undergraduate Berkeley Initiative in Soft Computing research group, and its development was guided by Sergio Guadarrama. The authors would like to thank the BISC group for welcoming undergraduates and fostering their interests.

⁸ Such a system would probably not be able to use a vector-space internal representation

References

1. Berry, M. W., Browne, M. (1999) *Understanding Search Engines: Mathematical Modeling and Text Retrieval (Software, Environments, Tools)* SIAM, Philadelphia
2. Bezdek, J. C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.
3. Brin and Page (1997) *The Anatomy of a Large-Scale Hypertextual Web Search Engine*
4. De Cock, M., Guadarrama, S., Nikraves, M. (2004) *Fuzzy Thesauri for and from the WWW*. Paper prepared for this book.
5. Haveliwala, Gionis, Klein, Indyk (2002) *Evaluating Strategies for Similarity Search on the Web*
6. Kamvar, Klein, Manning (2002) *Spectral Learning*
7. Kummamuru, Dhawale, Krishnapuram (2003) *Fuzzy Co-clustering of Documents and Keywords*
8. Nikraves, M., Takagi, T., Tajima, M., Shinmura, A., Ohgaya, R., Taniguchi, K., Kazuyosi, K., Fukano, K., Aizawa, A. (2003) *Web Intelligence: Conceptual-Based Model*. Internal report, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, Memorandum No. UCB/ERL M03/19