Hierarchical Information Clustering Using Ontology Languages

Travis D. Breaux
Department of Computer Science,
North Carolina State University
tdbreaux@ncsu.edu

Joel W. Reed
Computation Sciences and Engineering,
Oak Ridge National Laboratory,
reedjw@ornl.gov

Abstract

The tools to analyze and visualize information from multiple, inhomogeneous sources have traditionally relied on improvements in statistical methods. The results from statistical methods, however, overlook relevant semantic features present within natural language and text-based information. Emerging research in ontology languages (e.g. RDF, RDFS, SUO-KIF, and OWL) offers promising avenues for overcoming these limitations by leveraging existing and future libraries of meta-data and semantic mark-up. Using semantic features (e.g. hypernyms, meronyms, synonyms, etc.) encoded in ontology languages, methods such as keyword search and clustering can be augmented to analyze and visualize documents at conceptually higher levels. We present findings from a hierarchical clustering system modified for ontological indexing and run on a topic-centric test collection of documents each with fewer than 200 words. Our findings show that ontologies can impose a complete interpretation or subjective clustering onto a document set that is at least as good as meta-word search.

1. Introduction

With the Internet and World Wide Web came improved distribution and storage capabilities of information and an increase in the production and expansion of personal, commercial, and government online services. Recently, emerging wireless and remote access technologies are further increasing the ubiquity of network access and the size of information flows. For this reason, information retrieval (IR) tasks capable of identifying the most relevant information have continued to receive growing attention with ontologies offering potential new approaches by providing deeper interpretations into information.

In particular, categorization and search tasks that use statistical methods such as Latent Semantic Indexing [1] or the Vector Space Model [2] combined with hierarchical clustering have been successfully demonstrated in a number of IR systems. While clustering offers a unique improvement over conventional, uninformed keyword search, traditional clustering requires sufficiently large populations of words before exact word matches can be used to decide relatedness. A fundamental limitation in these methods includes word indexing that is missing important semantic relationships available in emerging ontologies. An ontology provides specific relationships between words that can serve as an interpretation in the clustering algorithm. The ontology can provide a single point-of-view or be combined with other ontologies to produce more complex views of the information not previously obtainable by traditional methods.

This paper begins with a background in clustering and ontologies. In describing ontologies, we also provide a brief overview of semantic features commonly supported in ontology languages. Following, we introduce our approach using a hierarchical clustering system combined with our own ontology formatted in an extended RDF/ RDFS. Finally, the results of our implementation included visualizations are presented and discussed followed by a review of related work.

2. Background

Statistical methods in text-based information analysis generally seek to uncover correlations among word frequencies in a collection of documents. Perhaps the most elementary approach, the *keyword search*, organizes documents by indexed words. Extensions to this method apply various algorithms that produce relational rank factors specific to features in the information domain or user context such as link relevance among web pages [3], or feature usage in software applications [4]. In applications with limited *a priori* domain knowledge, popular approaches include document clustering obtained by computing *relatedness scores* using a vector space model. These scores rank and relate documents by word frequencies within documents, commonly called the *bag of words*, and normalizing an overall document score across several documents in a collection. The *term frequency inverse document frequency* (*TFIDF*) is a well established

relatedness score. In addition, a minimum level of word filtering aimed at reducing word form complexity (e.g. noun and verb stemming, contraction expansion, etc.) such as Porter stemming [5] or by reducing the number of statistically irrelevant words known as *stop words* [6] (e.g., articles, pronouns, prepositions, etc.) is performed. The general theory behind relatedness scores in text-based information analysis follows: abstract concepts are largely represented by nouns (e.g. persons, places, or things) and verbs (e.g., actions and some events) and conceptually related documents will share similar nouns and verbs. The frequency of relatedness, therefore, attempts to describe "just how close" two documents are by counting the number of common nouns and verbs between them.

Present-day ontologies can be grouped into two general categories: those that form meta-language dictionaries and those that are derived from knowledge bases built for inference engines and expert systems. In the former group, the ontology is organized around the words in a natural language via their lexical attributes (i.e. part-of-speech) and semantic relations. In the latter group, the ontology is composed of predicates that in appearance are words or word phrases from natural language (e.g. FruitOrVegetable[1]) or concepts using several semantic relations (e.g., AboveGroundLevelInAConstruction[1]). Since the content of these ontologies primarily serves as logical predicates, there is little emphasis placed on explicitly encoding individual relations such as in the case of dictionary-style ontologies. In addition to the non-orthogonal conceptual predicates, the latter group often lacks verbs as another consequence of conventional formal inference (i.e., logical implications replacing terms indicative of state transitions.) For IR applications that primarily use the natural language content of documents in their sorting algorithms, the dictionary-based ontologies are best suited for expanding relationships between terms within a document.

2.1 Ontology Languages

Ontology languages provide the formal structures that link terms through semantic relations. The categorical, taxonomic or class relations for hypernyms (i.e., super-class) and hyponyms (i.e., sub-class) used in term abstraction and refinement, respectively, are so popular they almost uniquely define the ontological prospect in many applications. In natural language, these relations are applicable to both nouns and verbs, although, the emphasis in ontology development has been mostly on nouns or concepts that are compositions of several word forms (i.e., nouns, prepositions, verbs, etc.) The part-whole relations for meronyms (i.e., parts of a whole) and holonyms (i.e., whole of its parts) are perhaps the next most important ontological features for nouns. Unlike the categorical relations, the part-whole relations have a number of variations exclusive to certain nouns [7], complicating the separation of part-whole structure from content which is desirable in ontology language design. Other common noun-specific relations include synonyms, antonyms, and homonyms.

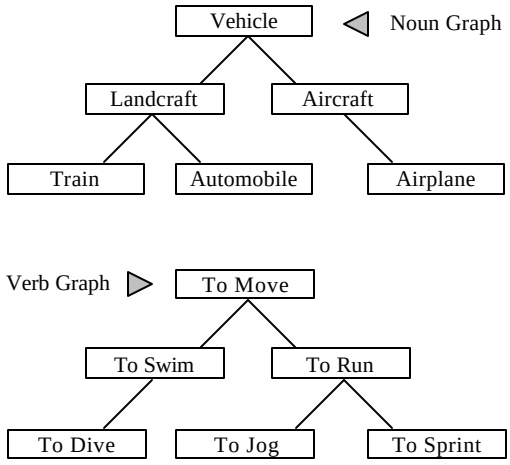
Exactly which relationships and other features are present in an ontology language is dictated by the intended application of a specific language. For example, the ontology languages based on subsets of first-order logic place more emphasis on logical operators and set-theoretic relations including disjointedness, transitivity and equivalence classes. Alternatively, part-whole relations are very popular in medical ontology languages where the need to describe the composition of biological systems is an obvious priority. Evaluating an ontology language is therefore a matter of determining what relationships are supported by the language and required by the ontology or application domain. Adapting an existing ontology to a new application requires the ability to distinguish and separate desirable features from the undesirable to guarantee both the quality and persistent availability of extracted information.

Ontology languages may be community standards, such as LOOM [8], or they may be unique to one implementation, such as Princeton University's WordNet [9]. Recently, there has been much effort to develop standard ontology mark-up languages for indexing and searching HTML documents. Simple HTML Ontology Extensions (SHOE) is an ontology language intended to provide inference capability over arbitrary categories, relations and custom data-types [10]. Publishers mark-up existing documents with SHOE instances, referencing external SHOE ontologies that either stand-alone or extend other ontologies. Developers of SHOE have since deferred their efforts to the Semantic Web. The Web Ontology Language (OWL) is a continuing W3C project derived from a number of efforts including RDF/ RDFS, the Semantic Web, DAML, and OIL. In the spirit of SHOE, OWL provides a language for composing ontologies that can be aligned with HTML content. Whereas the SHOE language implements a form of Horn logic, OWL attempts to implement Description Logic as an extension of RDF [11]. In both efforts, the formalism of the ontology language is driven by the desired inferential capabilities found in their respective logics. The inferential capability is added to the source documents and never extracted from the human-readable document content.

[1] Acquired from the Cyc Upper Ontology provided by Cycorp, Inc. as a contribution to the DAML project.

3. Approach/ Implementation

A complimentary approach to pure hierarchical clustering makes use of the classification hierarchy common to ontologies. Such hierarchies are typically terminated at their "roots" by the most general words (e.g., thing, entity, object, action) with the most specific words lacking obvious refinements. Figure 3.1 illustrates examples of the abstraction-refinement hierarchy for a few nouns and verbs. While nouns and verbs can be organized in these hierarchies, the apparent existence of multiple hypernyms for a single word dictates that these hierarchies are not simple, rooted trees as the figure might suggest. In addition to abstraction and refinement, the graph constructed from the hierarchy of nouns can also be extended with other semantic relationships for meronyms, synonyms, etc., which altogether form an ontology.



Figures 3.1: Examples of nouns and verbs in the "vehicle" and "to move" hierarchy, respectively, are shown. Downward arrows point from a conceptually abstract word toward word refinements. Searches among abstract words would also capture, semantically, documents that contain the refined words. Negation could be used to "trim" the search tree, e.g. find all of the documents within "to move" excluding those within "to swim."

In our approach, we combine ontologies encoded in an extended form of RDF/ RDFS with an established hierarchical clustering system. We chose RDF/ RDFS since it is an extension of XML and a World-Wide-Web Consortium (W3C) standard with syntactical support for defining a classification system. Another advantage is the many publicly available parsers for XML and their extensions. Languages such as LOOM and OWL introduced features unnecessary in our approach.

Our extensions to RDF include an equivalence relation and a class naming convention. In addition to the existing classification syntax for hypernyms, we've extended RDF to include an equivalence relation between classes to support the declaration of synonyms. The equivalence relation is similar to the *sameAs* relation defined in OWL and DAML+OIL. Since terms in our ontologies may span multiple words and RDF class names do not allow spaces, a naming convention has been adopted to resolve this inconsistency. The naming convention follows: insert a space between 1) any character followed by a capital letter and 2) any letter followed by a digit. Acronyms, therefore, are all lowercase, unless spaces between the letters are desirable. Punctuated class names such as hyphenated terms are not affected given they conform to this convention. Following is an example class for drone aircraft:

```
<rdfs:Class rdf:ID="UnmannedArialVehicle">
    <rdfs:subClassOf rdf:resource="#Aircraft"/>
    <rdfx:sameAs rdf:resource="#uav"/>
    <rdfx:sameAs rdf:resource="#DroneAircraft"/>
</rdfs:Class>
```

Figures 3.2: Sample from our extended RDFS ontology for the term "drone aircraft."

The semantics of our ontologies include additional constraints. All terms within our ontologies are nouns or proper names from the English language. Terms may have multiple hypernyms but cycles are not permitted. Root terms in the ontology have no hypernyms but may have zero or more hyponyms. Naturally, a root term with no hyponyms and no synonyms is acceptable but not very interesting. Homonyms, or terms with the same spelling but perceivably separate meanings, however are treated as a separate word-sense disambiguation problem and therefore were not permitted to appear either within the ontology or the document test collections.

For clustering we used a fully automated, hierarchical clustering system that has been rigorously tested on collections of text documents [12]. The system hierarchically clusters documents and produces rich visualizations in the form of non-rooted dendrograms. The procedure for adding text documents into the system involves parsing, filtering, and indexing terms into individual document vectors. Among other things, the process of parsing and filtering includes a stop-word list and a Porter stemmer. Each term is then indexed into a local frequency vector which maintains the collective term frequency for the originating document. After the document has been fully indexed, the local vector is merged into the global matrix which accounts for the term frequency across the entire collection of documents.

Together, these frequencies are used to calculate the term weights which comprise a normalized document vector. Finally, the dissimilarity matrix is built from the pair wise dot product of each document vector in the collection.

Our approach uses the ontology during the filtering phase of the document acquisition process, subsequent to the reduction of terms to a single synonym. For each reduced term within a document, a matching term is located within the ontology. If the term is matched, the transitive closure of the hypernym set for the matched term is then added to the document. This set is known to be finite since our ontologies do not permit cycles. If the term is not matched within the ontology, the term is removed from the document. The resulting effect clusters documents exclusively by the relationships shared between the ontology and the documents. Finally, the clustering threshold is maintained at 100% similarity producing the most refined clusters.

3.1 Comparing Approaches

The dissimilarity matrix is the source for constructing the hierarchical clusters and comparing the differences between one hierarchical clustering method and another. Building the hierarchical clusters proceeds from the following basic algorithm: 1) initially let each document represent a singleton cluster, 2) locate the two "nearest" clusters and create a new compound cluster with a new compound dissimilarity score, 3) repeat step two until either a) there is only one cluster remaining or b) each remaining cluster's dissimilarity score exceeds a constant threshold value. The final non-rooted dendrogram in the visualization is then built from either a) a single tree or b) a forest of trees, depending on the termination case of the algorithm. Comparing two clustering methods involves a pair wise comparison between sequences of document dissimilarity scores from one method to those of another. Following is our algorithm for determining the percentage difference between two hierarchical clustering methods:

- Let $X = \{ x_{1,2}, x_{1,3}, \ldots, x_{n,n-1} \}$ be the dissimilarity matrix X for some method with dissimilarity values $x_{i,j}$ for two different document indices i, j and let $\alpha(X, i, j) = x_{i,j}$ such that $x_{i,j} \in X$.

- Let $d(X, Y, \langle i, j, k \rangle)$ return 1 if the relationship between $x_{i,j}$ and $x_{j,k}$ is not maintained between $y_{i,j}$ and $y_{j,k}$ for two matrices X, Y, and return 0 otherwise. These cases are characterized below:

$$\begin{cases} 1, [\, \alpha(X,i,j) < \alpha(X,j,k) \,] \wedge [\, \alpha(Y,i,j) \geq \alpha(Y,j,k) \,] \\ 1, [\, \alpha(X,i,j) > \alpha(X,j,k) \,] \wedge [\, \alpha(Y,i,j) \leq \alpha(Y,j,k) \,] \\ 1, [\, \alpha(X,i,j) = \alpha(X,j,k) \,] \wedge [\, \alpha(Y,i,j) \neq \alpha(Y,j,k) \,] \\ 0, \textit{otherwise} \end{cases}$$

- Let $\gamma(X, Y)$ return the summation for relationships not maintained among unique triples $\langle i, j, k \rangle$ of document indices normalized by the total number of such triples (i.e., "n choose 3") in a collection of n documents. This value is the percentage difference between the matrices X, Y.

$$\gamma(X,Y) = \sum_{i=0}^{n-3} \sum_{j=i+1}^{n-2} \sum_{k=j+1}^{n-1} d(X,Y,\langle i,j,k \rangle) \Big/ \binom{n}{3}$$

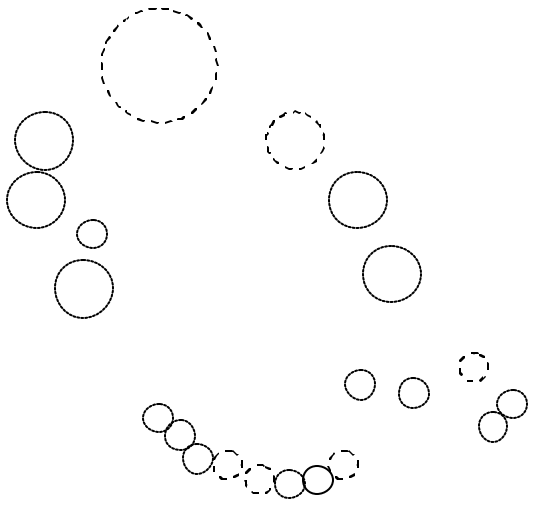
4. Results

The test collection includes 29 articles from various online news sources. Each article is between 150 and 200 words in length with the article subject matter concerning an event involving one or more types of vehicle. The following four titles are taken from documents in the test collection and provided as a brief overview:

- Raytheon awarded contract for new F/A-22 fighter
- Police motorbike stolen on surveillance operation
- Four Abu Sayyaf members killed in trawler encounter
- Japan, China plan first-ever mutual warship visits

The combined ontologies include a classification and synonym structure for types of vehicles with a total of 46 term classes including instances and 21 synonyms. The ontology is split among three sub-domains, one for aircraft, landcraft and watercraft with a separate upper ontology that bridges these three domains.

Running our system on the test collection without the ontology produces the baseline clustering image in figure 4.1, below. Each document was traced within the visualization to show the evident mix of documents and demonstrate the weakness of the vehicular organization in the baseline clustering hierarchy. The evident mix of documents is attributed to the document vectors that include significant interference from terms not represented by the ontology or the traced interpretation.



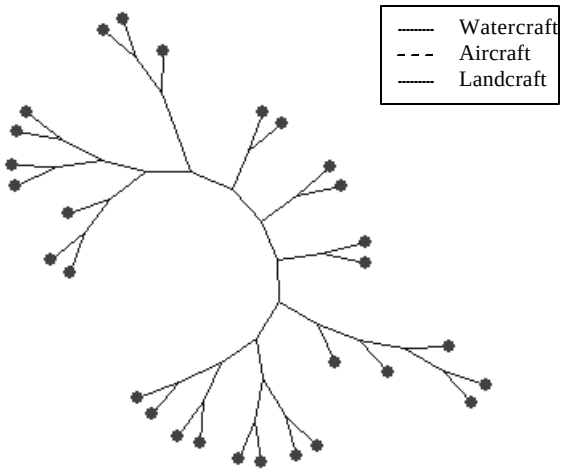


Figure 4.1: Before applying the ontology, the system run on the 29 documents in the test collection at a maximum 100% clustering threshold.

Running our system on the test collection with the ontology shows dramatically different results presented in figure 4.2, below. The resulting clusters significantly correspond to abstract terms within the ontology such as aircraft, automobile, and ship. In general, the clustering completely partitions the test collection into the three branches described by the given ontology. Furthermore, in both cases specific refinements were characterized by separate clusters. For example, all of the documents describing a warcraft in the aircraft and watercraft branches are exclusively found in unique warplane and warship clusters, respectively. The larger automobile cluster results from relatively less refinement in that part of the landcraft ontology.

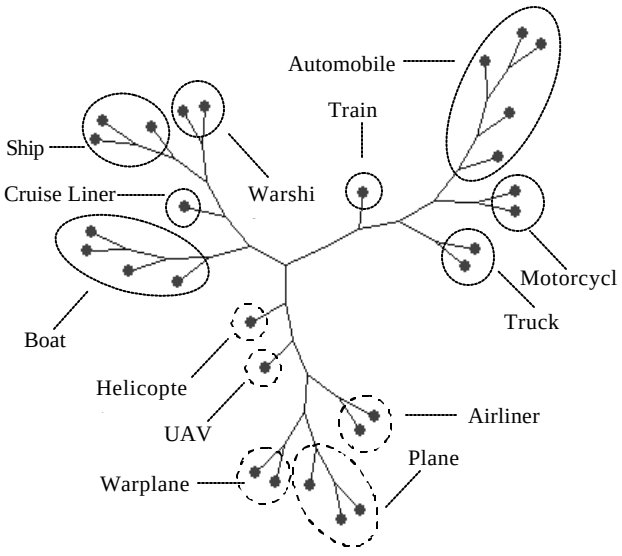


Figure 4.2: After applying the ontology with a 100% clustering threshold, the 29 documents cluster exclusively by vehicle type from three sub-domains: landcraft, watercraft, and aircraft. Notably, a greater level of refinement is present

Using the comparison function γ with the dissimilarity matrices that generated the visualizations in figure 4.1 and 4.2, the percentage difference between the clusterings was significant at 62.42 % – suggesting that almost two-thirds of the relationships between documents were inverted from figure 4.1 to produce the clustering in figure 4.2.

5. Discussion

Applying the ontologies to the filtering and indexing phase of clustering cleanly partitions the documents into disjoint, clustered branches with a bijection to the three sub-domains characterized in the combined ontology. While the results hold promise for applications of dictionary-based ontologies in information retrieval tasks, they also raise an important question: How can we quantify the significance of ontological clustering beyond the similar effects of the meta-word search?

Our results suggest the constraints imposed on our ontologies and our test collection significantly impacted the presence of the bijection. A similar bijection can be obtained by the method for meta-word search discussed earlier with figure 3.1. Each of the three major branches and many of the smaller sub-branches, likewise, characterize a simple meta-word search performed on the same number of documents minus the overhead of clustering. Regardless, these results do establish a baseline in which hierarchical clustering using ontologies is at least as good as meta-word search.

Beyond the baseline, however, clustering larger-size documents or using ontologies from multiple domains with varied semantic relations represent the frontier in ontological clustering. Questions along this frontier include:

- How do different domains and relationships in the ontology impact the significance and quality of clusters? Considering different relations such as place/ location, part/ whole, and capabilities of entities would produce far more complicated clustering results. Further analysis comparing the significance of these patterns with relations in a corresponding ontology requires the development of new semantic metrics that go beyond traditional statistical measures.
- What role does ontological clustering play in information extraction? The simple process of filtering and indexing documents by their ontological relationships prescribes structured

significance to the "meaning" of documents. While classification hierarchies only suggest "what a document is about," other relations and process-oriented knowledge assigns richer significance to documents. Clustering algorithms that rely solely on statistical correlations may only serve to disrupt the more complex semantic significance attributed to document collections by richer ontologies.

6. Related Work

Earlier work with hierarchical thesauri sets and query expansion examined ontological features in information retrieval. Hierarchical thesauri, like simple ontologies, attempt to categorize terms by their broader, more general synonyms and vice versa. In query expansion, an initial query is expanded to include information based on a particular heuristic. Using a heuristic that leverages hierarchical thesauri, the queries can be re-written to include specific terms not provided in the original query. Voorhees examined the application of the WordNet dictionary, in particular the classification hierarchy of nouns, to query expansion of topic statements, or complex natural language queries, in the Text Retrieval Conference (TREC) collections [13]. Voorhees found that WordNet synsets improved the results of simple queries with very few words but showed no improvement on larger queries. Voorhees results are significant since they demonstrate how terms expanded by hypernyms improve indexing on small words sets.

Hotho et al. demonstrate that using ontologies as filters in term selection prior to the application of a K-Means clustering algorithm will increase the tightness and relative isolation of document clusters as a measure of improvement [14]. K-Means clustering is a non-hierarchical method that establishes a fixed *k* number of clusters. Each document is then marshaled into a non-optimal cluster using a heuristic such as the sum of squared Euclidean distances from the mean of each cluster. The less optimal K-Means clustering is preferred for its speed over its loss of accuracy. The ontology used by Hotho et al. uses a custom ontology language and expresses a taxonomy of "concepts" similar to WordNet synsets. Our approach uses hierarchical clustering which we believe better retains information between documents than K-Means at an affordable computational cost.

Maedche and Zacharias examine hierarchical clustering of ontology-based metadata for the Semantic Web [15]. In clustering metadata, they introduce a number of semantic measures required to compute the similarity matrix prior to constructing the clusters. These measures compute relatedness scores based on the relational similarity of two concepts, such as comparing their locations in a classification hierarchy or evaluating the intersection of their attributes. Unlike their approach, our algorithm clusters text documents using expanded term sets derived from the ontologies. As a result, our approach avoids the complexity of comparing conceptual graphs. In addition, we are able to demonstrate the relative improvement of using ontological term expansion over traditional hierarchical clustering that does not use ontologies.

7. Conclusion

It has been demonstrated that combining hierarchical clustering with ontologies provides significant advantages over traditional, non-ontological clustering. Using a test collection of documents with less than 200 words per document, our approach imposes a subjective view onto the resulting clusters driven by the content and organization of the ontologies. The statistical and visual significance of the differences between our approach and traditional, non-ontological clustering was presented using a mathematical dissimilarity measure and non-rooted dendrograms, respectively. Finally, our results show that hierarchical clustering using term expansion is at least as good as meta-word search.

Future work requires new methods for more complex analysis comparing clusters to relationships maintained within richer ontologies. Such methods must include relationships beyond simple classification hierarchies of terms, such as part/ whole, location/ place, and capabilities of entities. It has yet to be determined if these extended features can contribute to the evaluation of cluster quality or whether they will primarily be used in more complex information extraction tasks.

8. References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis" *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391-407, 1990.

[2] G. Salton, M. Lesk, "Computer Evaluation of Indexing and Text Processing", Journal of the ACM, vol. 15, no. 1, pp. 8-36, 1968.

[3] D. Heckerman, E. Horvitz, "Inferring Informational Goals from Free-Text Queries: A Bayesian Approach." In Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98), San Francisco, CA, USA, pp. 230-237, 1998.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web.” Stanford Digital Libraries Working Paper, 1998.

[5] M. E. Winston, R. Chaffin, and D. Hermann, “A Taxonomy of Part-Whole Relations.” Cognitive Science, vol. 11, pp. 417-444, 1987.

[6] M. Porter, “An algorithm for suffix stripping.” Program, vol. 14, no. 3, pp. 130-137, 1980.

[7] C. Fox, “Lexical analysis and stoplists.” In Information Retrieval: Data Structures and Algorithms (ed. W.B. Frakes and R. Baeza-Yates), Englewood Cliffs, NJ: Prentice Hall, 1992.

[8] R. M. MacGregor, “A Deductive Pattern Matcher.” In Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI ’88), pp. 403-408, 1988.

[9] G. A. Miller, “WordNet: a dictionary browser.” In Proceedings of the First International Conference on Information in Data, University of Waterloo, Waterloo, 1985.

[10] J. Heflin, J. Hendler, and S. Luke, “Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web.” In AI and Information Integration. Papers from the 1998 Workshop. WS-98-14. AAAI Press, 1998. pp. 51-57.

[11] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, "From SHIQ and RDF to OWL: The making of a web ontology language.” Journal of Web Semantics, vol. 1, no. 1, pp. 7-26, 2003.

[12] T. E. Potok, M. Elmore, J. Reed, F. T. Sheldon, “VIPAR: Advanced Information Agents Discovering Knowledge in an open and changing environment.” In Proceedings of the 7th World Multi-conference on Systemics, Cybernetics, and Informatics Special Session on Agent-Based Computing, Orlando, FL, USA, pp. 28-33, 2003.

[13] E. M. Vorhees, “Query Expansion using Lexical-Semantic Relations.” In Proceedings of the 17th International Conference on Research and Development in Information Retrieval, Dublin, Ireland, pp. 61-69, 1994.

[14] A. Hotho, S. Staab, G. Stumme, “Ontologies Improve Text Document Clustering.” In Proceedings of the 3rd IEEE Conference on Data Mining, Melbourne, FL, USA, pp. 541-544, 2003.

[15] A. Maedche and V. Zacharias, “Clustering Ontology-based Metadata in the Semantic Web.” In Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), Helsinki, Finland, pp. 342-360, 2002.