Ontology-based Semantic Classification of Unstructured Documents

Ching Kang Cheng[1], Xiao Shan Pan[2], Franz Kurfess[1]

[1]Department of Computer Science California Polytechnic State University
San Luis Obispo, California 93407, USA
{fkurfess, ckcheng}@calpoly.edu

[2]Department of Civil and Environmental Engineering Stanford University
Stanford, California 94305, USA
xpan@stanford.edu

Abstract. As more and more knowledge and information becomes available through computers, a critical capability of systems supporting knowledge management is the classification of documents into categories that are meaningful to the user. In a step beyond the use of keywords, we developed a system that analyzes the sentences contained in unstructured or semi-structured documents, and utilizes an ontology reflecting the domain knowledge for a semantic classification of the documents. An experimental system has been implemented for the analysis of small documents in combination with a limited ontology; an extension to larger sets of documents and extended ontologies, together with an application to practical tasks, is the focus of ongoing work.

1.0 Introduction

With the volume of knowledge and information available to computer users increasing at an ever accelerating rate, the need for an effective mechanism to organize not only information, but also knowledge becomes critically important. Document clustering techniques have been employed frequently to support the organization and retrieval of information [1]. Information retrieval, however, leaves a significant portion of the utilization of knowledge contained in the retrieved documents to the user: Typically, these retrieval techniques are used to calculate a ranking of the documents, attempting to identify the ones that are most relevant to the user.

Document clustering is essentially an unsupervised process where a large collection of text document is organized into groups of documents that are related, without depending on external knowledge. A potential problem with the data-driven clustering algorithms is the inability to correctly identify cases when different words are used to describe the same concept. This is due to the similarity-based measure adopted in the algorithm. Furthermore, without including the user context, more often than not, information is organized according to the fixed viewpoint of the conventional clustering methods, rather than reflecting the interests of the users [1]. This will ultimately discount the usefulness of the information.

The core principle of our approach is based on our belief that knowledge and information has to be organized in a manner that is intuitive to the user. We have developed the OSC (Ontology-based Semantic Classification) framework, leveraging

on natural language processing techniques and ontologies to incorporate the user's current context into the categorization of information. Figure 1.0 illustrates the overall process where unstructured documents are categorized according to the user perspective. In Section 2, we discuss the usage of ontology in the OSC framework. Section 3 presents the various components employed. In Section 4, we show the implementation and in Section 5, we summarize our findings and future endeavor.

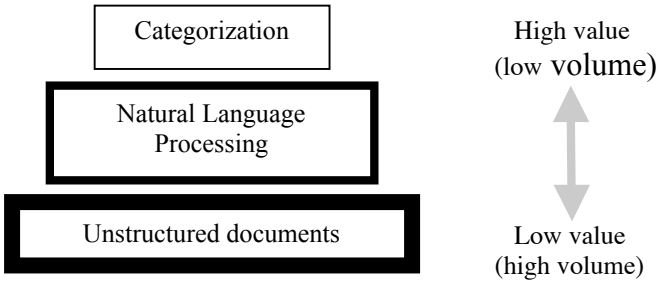

Fig. 1. Overall classification process

2.0 Ontology

Ontology construction is an approach to utilize computers for the structured representation of domain knowledge. An ontology can be defined as specification of a representational vocabulary for a shared domain of discourse which may include definitions of classes, relations, functions and other objects [2]. Ontology-based computer systems do not interact directly with the real world but rather with internal models of the relationships between concepts and objects in the real world. Such models represent problem domains, and the development of such models in computers is referred to as ontology building.

An ontology includes a selection of specific sets of vocabulary for domain knowledge model construction, and the context of each vocabulary is represented and constrained by the ontology. Therefore, an ontological model can effectively disambiguate meanings of words from free text sentences, overcoming the problem faced in natural language where a word may have multiple meanings depending on the applicable context [3].

Concepts represented by an ontology can usually be clearly depicted by a natural language because the ontology and the natural language function similarly (i.e., describing the world). Most vocabularies used in ontologies are direct subsets of natural languages. For example, a general ontology uses "thing", "entity", and "physical"; a specific ontology uses "BMW", "basketball", and "tree". Depending on the construction of the ontology, the meaning of those words in the ontology could remain the same as in natural language, or vary completely. The meaning of ontological terms that are not derived directly from a natural language can still be captured by a natural language. For example, the word "COM" used in a specific ontology means "Common Object Model" in English.

From an engineering perspective, ontologies can be very helpful with the reuse of domain knowledge, and for the separation of domain knowledge and software code that performs operations on that knowledge. We have adopted ontologies as the link to incorporate user-specific context into the categorization process within the framework. An ontology is used in both the CFTI (context-based free text interpreter) and the CCA (context-based categorization agent) parts of the framework

3.0 Semantic Classification

Linguistically, humans combine understanding of relatively small textual units in order to understand larger textual units, guided by syntactic and semantic rules. Syntax relates to arrangement, and semantic to the meaning of words. Similarly, it is necessary for a natural language processing system to be able to address syntactic and semantic aspects of natural language [3]. Subsequently, to perform useful classification, the categorization must be based on the actual information content or explicit representation of the information content of the source documents. The classification criteria must reflect the interest of the users. In this section, we introduce two existing language tools (i.e., Link Grammar Parser and WordNet), and the design of CFTI and CCA.

3.1 Syntactic Analysis

Natural language syntax affects the meaning of words and sentences. The very same words can have different meanings when arranged differently. For example: “a woman, without her man, is nothing” and “a woman: without her, man is nothing” (http://www.p6c.com/joke of the week.html). The Link Grammar Parser was found to be a very effective syntactic parser, and is therefore incorporated into the design of the CFTI.

3.1.1 Functions of Link Grammar Parser The Link Grammar Parser, developed at Carnegie Mellon University, is based on “link grammars”, an original theory of English syntax [4]. The parser assigns to a given sentence a valid syntactic structure, which consists of a set of labeled links connecting pairs of words.

The Link Grammar Parser utilizes a dictionary of approximately 60,000 word forms, which comprises a significant variety of syntactic constructions, including many considered rare or idiomatic. The parser is robust; it can disregard unrecognizable portions of sentences, and assign structures to recognized portions. It is able to intelligently guess, from context and spelling, probable syntactic categories of unknown words. It has knowledge of capitalization, numeric expressions, and a variety of punctuation symbols.

3.1.2 Basic Concepts of Link Grammar The basis of the theory of Link Grammar is planarity, described by [5], as a phenomenon evident in most sentences of most natural languages. To represent a sentence, arcs are drawn connecting words with specified relationships within sentences. These arcs do not cross for syntactically

correct sentences. Planarity is defined in Link Grammar as "the links are drawn above the sentence and do not cross" [4]. To visualize link grammars, think of words as blocks with connectors coming out. There are different types of connectors; connectors may also point to the right or to the left. A sentence is valid if all the words present are used according to their rules, and certain global rules are satisfied [6]. Each word, from a Link Grammar perspective, is a block with connectors (see Figure 2).

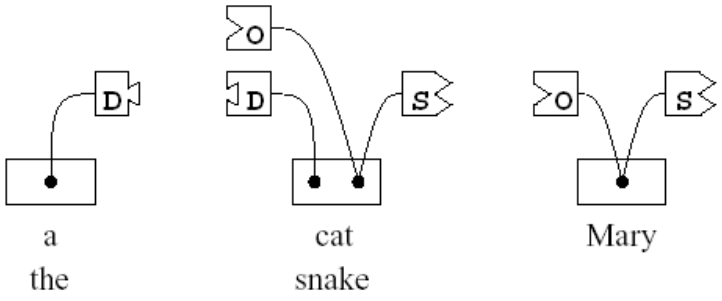

Fig. 2. Each word is a block with connectors [6].

Each intricately shaped, labeled box is a connector. A connector is 'satisfied' when 'plugged into' a compatible connector (as indicated by shape). A valid sentence is one in which all blocks are connected without a crossing. An example of a valid sentence is "the cat chased a snake" (Figure 3).

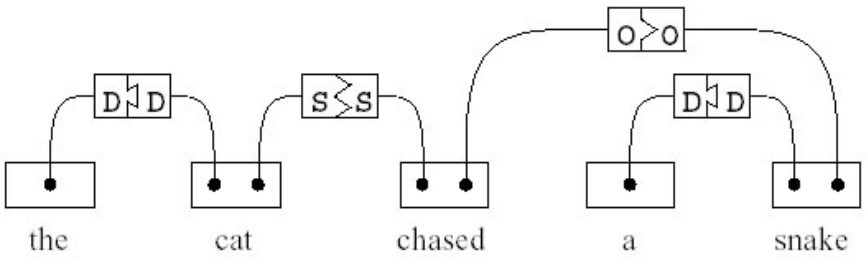

Fig. 3. A valid sentence contains blocks connected without a cross [6].

An example of an invalid sentence is "the Mary chased cat", which contains a cross (Figure 4).

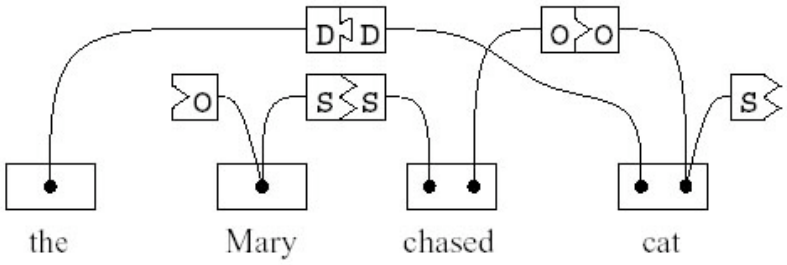

Fig. 4. An invalid sentence contains blocks connected with crosses [6].

The Link Grammar Parser identifies all valid linkages within a free text input, and outputs them as grammatical tree. For example, an input such as "The brown fox jumped over that lazy dog" would result in the output shown in Figure 5:

```
+-----Ds-----+              |     +-----Dsu----+
|     +---A--+---Ss--+--MVp-+     |      +--A--+
|     |      |       |      |     |      |     |
the brown.a fox.n jumped.v over that.d lazy.a dog.n

Constituent tree:
(S (NP The brown fox)
   (VP jumped
       (PP over
           (NP that lazy dog)))
   .)
```

Fig. 5. An output produced by the Link Grammar Parser.

3.2 Semantic Knowledge

Two types of semantic knowledge are essential in a natural language processing system:

1. lexical knowledge among words independent of context (e.g., "children" as the plural form of "child", and the synonym relationship between "helicopter" and "whirlybird")
2. contextual knowledge (i.e., how meanings are refined when used in a specified context)

In CFTI, lexical knowledge is acquired through integration of the system with the WordNet database, and contextual knowledge is acquired by tracking contextual meanings of words and phrases during and after development of an ontology (i.e., context model).

3.2.1 WordNet Database WordNet, an electronic lexical database, is considered to be the most important resource available to researchers in computational linguistics, text analysis, and many related areas [7].

WordNet has been under development since 1985 by the Cognitive Science Laboratory at Princeton University under the direction of Professor George A. Miller. Its design is "…inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, and adjectives are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets." [8]

The most basic semantic relationship in WordNet is synonymy. Sets of synonyms, referred to as synsets, form the basic building blocks. Each synset has a unique identifier (ID), a specific definition, and relationships (e.g., inheritance, composition, entailment, etc.) with other synsets.

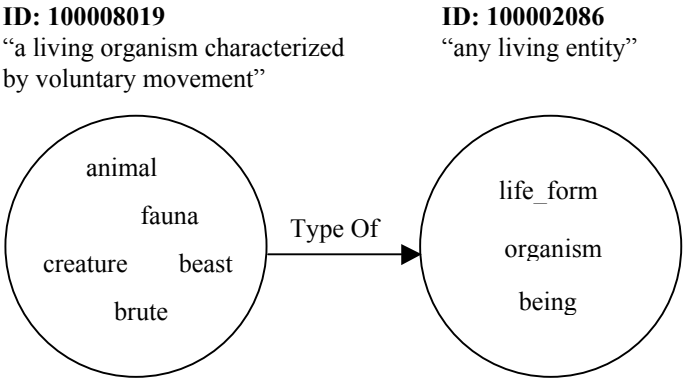

Fig. 6. Two synsets with a 'type-of' relationship.

Two synsets with a "type-of" relationship are shown in Figure 6. The first synset has an ID "100008019", a definition of "a living organism characterized by voluntary movement", and contains six individual words (e.g., "animal", "animate being", etc.). The second synset has an ID "100002086", a definition of "any living entity", and it contains three words (e.g., "life form", "organism", and "being"). The first synset is a "type-of" the second synset.

WordNet contains a significant amount of information about the English language. It provides meanings of individual words (as does a traditional dictionary), and also provides relationships among words. The latter is particularly useful in linguistic computing.

While WordNet links words and concepts through a variety of semantic relationships based on similarity and contrast, it "does not give any information about the context in which the word forms and senses occur" [7]. In CFTI, refinement of word meanings in specific contexts (i.e., contextual knowledge) is accomplished by mapping relationships between natural language and a context model.

3.2.2 Relationships Between Natural Language and Context Model Ontologies provide context for vocabularies which they contain. Direct and indirect mapping

relationships exist among ontological vocabularies and natural language vocabularies. Understanding of such relationships may enable a system to understand contextual meanings of words used in the context defined by an ontology. The application of the same word to other ontologies could produce other meanings [3].

In practice, the tracking of mapped relationships between a natural language sentence and a context model is a process of interpretation of the model (i.e., what a model really means) through the use of a natural language. Contextual knowledge can be attained directly from the ontology designer, or can be attained through utilization of an automated process if the ontology design follows formalized conventions.

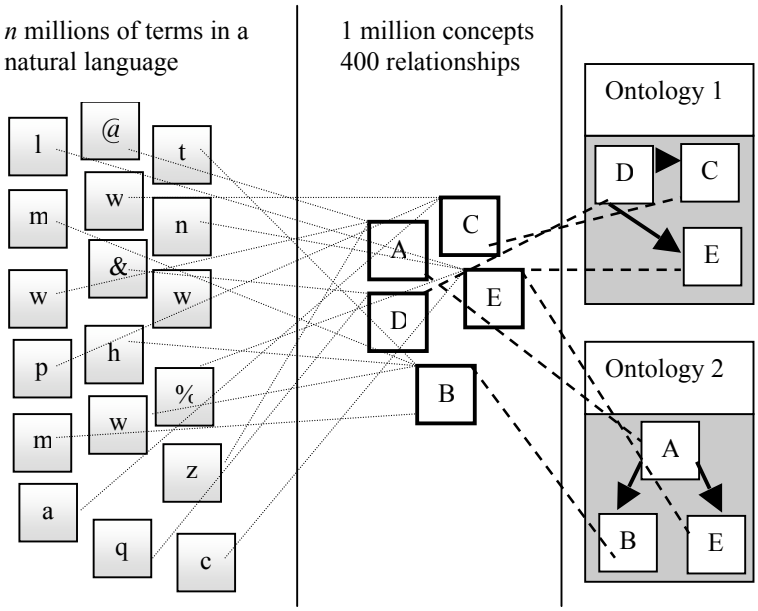

Fig. 7. Mapping from natural language to context models.

From the perspective of a natural language processing system which employs appropriate lexical and contextual knowledge, the interpretation of a free text sentence is a process of mapping the sentence from natural language to a context model (Figure 7). Different context models may produce different results simply because words can have different meanings in different contexts.

3.2.3 Representation of Meaning Understanding a free text sentence is a process of representing the sentence's meaning through the use of a model internal to an interpreter. This concept is applicable to both humans and computers. In CFTI, the representation of meaning is accomplished by manipulations of a context model (i.e., creation, modification, and deletion of objects and relationships in an object model).

For example, a hazard detection system receives a free text sentence "House 303 is on fire!". If the system is able to model this information correctly (i.e., locate the

instance of the house in the model and set its attribute to "on fire"), then it is assumed that the system understands the meaning of the sentence [3].

3.3 Context-Based Free Text Interpreter (CFTI) Design

CFTI leverages on the Link Grammar capability for syntactical analysis of a sentence. At the same time, the lexical meaning analysis of a sentence is supported through the integration with the WordNet database [3]. The tasks performed by CFTI are summarized as follows:

1). Analyze the syntactic structure of the sentence.
2). Analyze the lexical meaning of the words in the sentence.
3). Refine the meanings of the words through the application of a context model.
4). Represent the meaning of the sentence in the model.

Figure 8 illustrates the processing of a free text message by the CFTI system and the subsequent representation in the model.

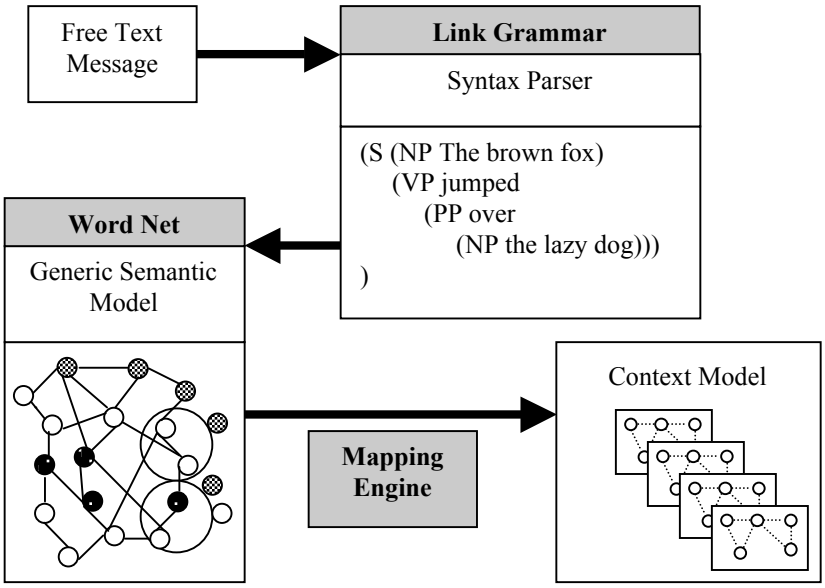

Fig. 8. From free text messages to context models.

Even though the CFTI requires an ontological model for the acquisition of contextual knowledge and the representation of meanings, the system is not constrained by any particular knowledge domain. A system change from one ontological model to another does not require significant system reconfigurations.

3.4 Context-Based Categorization Agent (CCA) Design

The context models produced from CFTI correlate the content of a particular document with the context of the user. The role of the CCA is to further enhance the usability of these context models by classifying them according to the user interest.

CCA relies on the ontologies to incorporate the category knowledge specified by the user. A key feature of such an approach is the capability to extend the ontology to include new knowledge without recompilation of the CCA. For example, a new category can be added to the ontology easily without having to re-configure CCA. CAA dynamically includes this new category in the categorization process.

The tasks performed by CAA are:

1). Interface with the context models.

2). Interface with the category ontology.

3). Classify the context models through the application of a category ontology.

4). Represent the classification of the document in the model.

Figure 9 illustrates the classification of the context models by the CAA and the subsequent representation in the model.

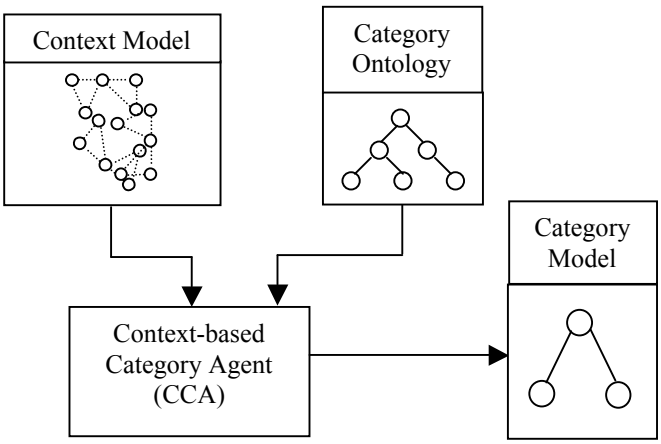

Fig. 9. From context models to category models

4.0 Implementation

This section explains a prototypical implementation of an ontology-based system for the semantic classification of unstructured documents. We demonstrate the feasibility of incorporating user context for the task of classifying unstructured documents. But first, we present the architecture of the overall framework.

4.1 Ontology-based Semantic Classification (OSC) Framework

The core design principle of the OSC framework is to provide loosely coupled yet seamlessly integrated components. To achieve this, the OSC framework architecture is

decomposed into three distinct layers and the interfaces between the components are specified in a language neutral format (e.g. via XML), as shown in Figure 10.

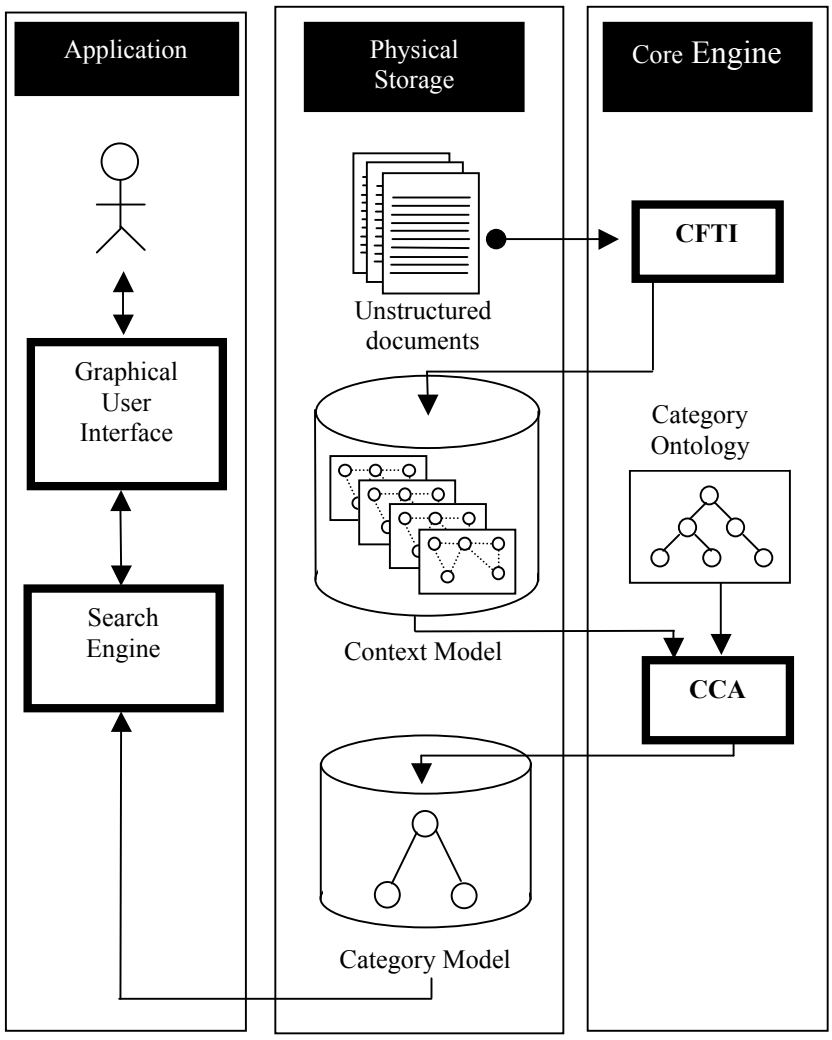

Fig. 10. System Architecture of the OKM Framework

4.1.1 Core Engine Layer The core engine layer encompasses components that contribute to the core functionality of the framework. It includes the CFTI and CCA.

CFTI is implemented through the use of CLIPS 6.20. The system contains five components: Link Grammar, Lisp Simulator, WordNet, a mapping engine, and a context model. The Link Grammar and Lisp Simulator process syntactic knowledge; WordNet provides lexical knowledge about words; the mapping engine is composed of CLIPS rules for meaning extraction from free text sentences; and the context model provides contextual knowledge about words and representation of meanings of free text

sentences [3]. While a context model is required by the system, a change from one context model to another does not require significant system reconfiguration.

CCA has been developed in CLIPS 6.20. It includes two components: a classification engine and a category ontology. The classification engine is powered by a network of rules that categorizes the context model with respect to the interest of the user as specified in the category ontology. The category ontology can be extended dynamically to allow changes without recompiling the system.

4.1.2 Physical Storage Layer The physical storage layer handles the storing of the context models, category models and the unstructured documents. The interface between the physical layer and the rest of the components is confined to a language neutral format such as XML, ensuring the loose coupling between the different layers. Applications that have to interact with the physical layer can be written in any programming language as long as that language supports XML. On the other hand, the context models, category models and the unstructured documents can be stored in text file format, binary file format, relational database and object oriented database.

4.1.3 Application Layer The application layer is a logical grouping of components that capitalize on the category models. By design of the OSC framework, application components can be plugged into the framework as and when they are ready.

A possible application component is the search engine. The search engine allows the users to query the category models using the context ontology as search criteria. We believe that the search is more accurate and helpful when the user can relate the search category to the domain knowledge. This is possible as the search category is created and maintained with the user's participation.

5.0 Conclusion

In this paper, we have shown how to include user context and preferences in the form of an ontology in order to classify unstructured documents into useful categories. We have demonstrated the use of a context-based free text interpreter (CFTI), which performs syntactical analysis and lexical semantic processing of sentences, to derive a description of the content of the unstructured document, with relevance to the context of the user.

Direct and indirect mapping relationships exist among vocabularies used by ontologies and vocabularies used by natural languages. The capture and utilization of these relationships is key to the development of natural language processing systems. The quality of classification of unstructured document is strongly dependent on the quality of context models and the accuracy of the interpretation of natural language.

The OSC framework has been tested with a relatively small-sized context model. While an assumption that the system would perform similarly when tested with larger-sized models seems valid, conducting such tests is the focus of ongoing work, together with the use of the OSC in practical applications.

References

1. Kim, H.J., Lee S.G.: A semi-supervised document clustering technique for information organization. Proceedings of the ninth international conference on Information and knowledge management. McLean, Virginia (2000)
2. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition, An International Journal of Knowledge Acquisition for Knowledge-Based Systems, 5(2). June (1993)
3. Pan, X.S.: A context-based free text interpreter. California Polytechnic State University San Luis Obispo Master's Thesis - Computer Science Department. Aug (2002)
4. Sleator, D., and Temperley, D.: Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196. (1991)
5. Melcuk, I.: Dependency Syntax: Theory and Practice. New York: State University of New York Press. (1988)
6. Temperley, D., Sleator, D., and Lafferty, J.: An Introduction to the Link Grammar Parser. Technical report, Available: http://www.link.cs.cmu.edu/link/ March 1999
7. Fellbaum, C.: WordNet: An Electronic Lexical Database. Cambridge: MIT Press. (1999)
8. Miller, G.: Wordnet: An Online Lexical Database. Int'l J. Lexicography, Vol. 3, No. 4, 1990, pp. 235-312. (1990)