Fall 2004

Sense -based text classification by semantic hierarchy representation

Xiaogang Peng
Louisiana Tech University

SENSE-BASED TEXT CLASSIFICATION BY SEMANTIC HIERARCHY REPRESENTATION

by

Xiaogang Peng, M.S.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

November 2004

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

07/29/2004
Date

We hereby recommend that the dissertation prepared under our supervision by XIAOGANG PENG

entitled Sense Based Text Classification By Semantic Hierarchy Representation

be accepted in partial fulfillment of the requirements for the Degree of

Ph.D. in Computational Analysis and Modeling

Supervisor of Dissertation Research

Richard J. Greechie
Head of Department

Department

Recommendation concurred in:

Weizhong Dai

Advisory Committee

Approved:
Director of Graduate Studies

Dean of the College

Approved:
Dean of the Graduate School

GS Form 13
(5/03)

ABSTRACT

Automatic classification of web pages is an effective way to facilitate the process of retrieving information from the Internet. Currently, two major classification methods are used in this area: keyword-based classification and sense-based classification. For keyword-based classification, keywords often have different semantic meanings, and the correct keyword matching is largely based on using exactly the same keywords. Thus, the classification results of keyword-based classification are not always satisfying. Many sense-based classification algorithms and systems have been presented, but they pay little attention to the relationship between senses. In this dissertation, we present a method to automatically classify documents based on the meanings of words and the relationships between groups of meanings or concepts. The classification algorithm builds on the word sense structures provided by a lexical database, which not only arranges words into groups of synonyms, but also arranges these groups of synonyms into hierarchies that represent the relationships between concepts.

Another problem with current classification systems is that most of them ignore the conflict between the fixed number of categories and the growing number of documents being added to the system. To address this problem, a category-based clustering method is developed to automatically extract a new category from a category that needs to be split. A category may be divided when the number of documents in the category is larger than a predefined size.

Experimental results show that the semantic hierarchy classification algorithm increases the classification accuracy by 13% compared to existing sense-based classification algorithms. The category-based clustering algorithm achieves a higher quality cluster than other existing methods that do not use category information. Combining the automatic classification based on word meanings and the dynamic addition of new categories based on clustering, we develop a new system to meet the current and future needs of a growing Internet.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author ____________________

Date 07/29/04

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ACKNOWLEDGMENTS

My sincere gratitude and appreciation goes first to my advisor, Dr. Ben Choi, for providing me with the unique opportunity to work in the research area of semantic web and web classification, for his expert guidance and mentorship, and for his encouragement and support at all levels. I would also like to thank Dr. Chris Cunningham, Dr. Weizhong Dai, and Dr. Galen Turner for their participation as Advisory Committee members and their valuable suggestions. Working environment is important to me, and I would like to thank all the members of the FIT team for their help and for being great team members.

Thanks to the Center for Entrepreneurship and Information Technology of Louisiana Tech University for research funding.

I also would like to express my gratitude to my parents, Haike Peng and Shaofan Chen, for the endless love and consideration they have always provided me. I would like to thank Yanni Chen for being a supportive wife. Finally, my silent thanks go to all the people and resources that contributed to my dissertation.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Today, the World Wide Web is growing rapidly [Galena 2000], but most Internet documents do not have a logical organization [Prasad 1999], which inevitably makes retrieving information difficult considering the number of documents on the web. The need for a fast way to select information in which we are interested becomes increasingly urgent. Assistance in retrieving documents on the web is provided by two kinds of tools: search engines and classified directories [Chandra et al. 1997].

Search engines allow keyword-based searches on the content of large collections of web documents. The weak points of current search engines are that they support only keyword search and the search returns a list of pages that includes a given set of keywords (or phrases). Most queries return a long list of pages, most of which are irrelevant and all of which include the given keywords. Some search engines, such as Yahoo [Yahoo] and Google [Google], offer "advanced search" tools to their users, yet the precision rate of these advanced searches is still not satisfying as reported in USA Today [2002].

In order to address problems, Yahoo [Yahoo] and Lycos [Lycos] use a manual classified directories method which organizes web pages into a category tree structure. A search using a classified directory is very convenient and usually leads the user to the set of documents he or she is seeking, but existing classified directories cover only a small fraction of the web. This limited coverage stems from the slow rate of web page classification by human labor.

Current manual classification of web pages, such as the one used by Yahoo, is not able to keep up with the rapid growth of the Internet. First, manual classification is slow and costly as it relies on skilled manpower. Second, the consistency of categorization is hard to maintain, as different people might have different classification standards based on their own experiences. Finally, the task of defining the categories is difficult and subjective, as new categories emerge continuously from many domains. Considering all these problems, the need for automatic classification becomes increasingly important.

Automatic text document classification is the task of assigning a text document to the most relevant category or several relevant categories by using computers. Formally, as found in Choi and Yao [2004], let C = $\{c_1, \ldots, c_m\}$ be a set of predefined categories, and D = $\{d_1, \ldots, d_n\}$ be a set of text documents that need to be classified. The task of text document classification is then transformed to approximate an unknown assignment function that maps $D \times C$ to a set of real numbers. Each number in the set is a measure representing the relationship of the document to the category and is used to determine the related categories for the document. A text document might belong to more than one category, depending on the definition and distinction of the category information.

In order for a machine to calculate the measure value for the relationship of a document to a category, the document and category should be represented in a machine-comprehensible format. This format is called *document representation*. Choosing the right document representation is one of the most important issues in text classification, because other operations, such as text learning and classification algorithm, are developed based on the representation. The *bag-of-words* representation in Koller and Sahami [1998] and Liang [1995] is a document representation that represents a document in a vector form. Each object in a vector is a word taken from the document along with the number of occurrences of the word in the document. This document representation is simple yet limited because it uses a word as the basic unit. Many experiments have been done to improve the performance by using a better document representation. For example, Mladenic [1998] extends the bag-of-words to the *bag-of-phrases* representation, which uses word sequences instead of single words as the basic unit. Chan [1999] also suggests that using phrases is a better choice than using words.

The bag-of-words or bag-of-phrases representation has two major problems. The first problem is that it counts word occurrence and fails to consider the fact that a word may have different meanings (or senses) in different documents or even in the same document. For example, the word "bank" may have at least two different senses, as in the "Bank" of America or the "bank" of the Mississippi River. However, using a bag-of-words representation, these two instances of "bank" are treated as if they are the same word. The second major problem lies in the fact that, occasionally, related documents may not share the same keywords, so those two related documents cannot be recognized as belonging to the same category.

The idea of changing a basic unit from word spelling to word meaning opens a new area in text classification, which is sense-based text classification. Recently, many sense-based text classification methods, as seen in Scott and Matwin [1998], Hsu and

Lang [1999] and Attardi et al. [1999], have been implemented, yet these methods do not make the most use of the semantic relations between the senses from a document.

Another problem with current classification systems is that most of them ignore the conflict between the fixed number of categories and the growing number of documents being added to the system. Most of the existing classification systems today put all their efforts on the document representation and classification algorithm in order to improve the accuracy of the classification and ignore the fact that the setting of predefined categories will also affect the classification performance. As the number of documents that need to be classified and stored in a defined category grows, the diversity of the documents will inevitably cause the original category to expand into subcategories more clearly defined for those documents. Generating additional categories in a predefined category hierarchy is called *category expansion*.

1.2 Contributions

In this dissertation, we present a new sense-based classification called a *semantic hierarchy classification system*. We suggest that the structure of the semantic relationships between senses is an important issue in sense-based classification and present a new *semantic hierarchy representation* (SHR) to describe the category and the documents that need to be classified. The document representation not only arranges keywords of a document into groups of synonyms, but also arranges these groups of synonyms into hierarchies that represent the relationships between concepts.

The system is also capable of creating new categories to solve the conflict between the fixed number of categories and the growing number of documents being added to the system with the help of the *category-based clustering* method. Different from normal clustering, we use a new measure, which is called the *category-based clustering score*, to describe the relationship between every pair of documents in a category needing expansion. The measure considers the similarity between two documents as well as the similarities of each of the documents to the category information.

The category-based clustering method provides a new way of combining text classification and clustering, which are tightly related in the information retrieval area. This method changes the idea of using clustering before classification and indicates that classification can also help clustering for some special purposes.

1.3 System Overview

This classification system can be divided into three parts (Figure 1.1): category description construction (Chapter 3), document classification (Chapter 4), and category expansion (Chapter 5).

In category description construction, we use the senses of category names and enrich them with semantically related senses in WordNet, which is a lexicon database providing sense mapping for words as well as semantic relations between senses in a tree structure. The keywords in the explanations of these senses are also used for the category description by turning them into senses. We assign a probability to each of the senses mentioned above and link these senses by using a recursive function to propagate the probabilities from the leaf node to the root, capturing the semantic relations between the senses. The resulting distribution of the probabilities of senses forms the semantic hierarchy representation. The last step of this process is capturing the hierarchy information of the category by applying the propagation function on the category hierarchy.

In document classification, we present a new text learning method to extract keywords from each of the documents that needs to be classified. Then, keywords are mapped to senses with the help of WordNet. After choosing the right sense for each of the keywords and calculating the probability of each sense in a document, we convert each of the documents to the semantic hierarchy representation format and adapt a classification algorithm based on the document representation to assign each document to a category in the predefined category hierarchy.

In category expansion, we calculate the category-based clustering score for each pair of documents in a category that needs to be expanded. We treat this score as an edge between two nodes representing the two documents. Then, a maximum spanning tree algorithm is applied on all these nodes to form a cluster, which is considered to be the new category.

1.4 Organization

The remainder of this dissertation is structured as follows: In Chapter 2, we outline related techniques that have been previously used. Then, we present the semantic hierarchy classification system in detail in the subsequent three chapters. The category description construction part is discussed in Chapter 3. In Chapter 4, we describe sense-based document classification algorithm. In Chapter 5, we continue discussing category expansion and describe a category-based clustering method for this purpose. Then, in Chapter 6, we provide the testing and performance analysis for our sense-based classification system. Finally, we provide a conclusion and future research directions in Chapter 7.

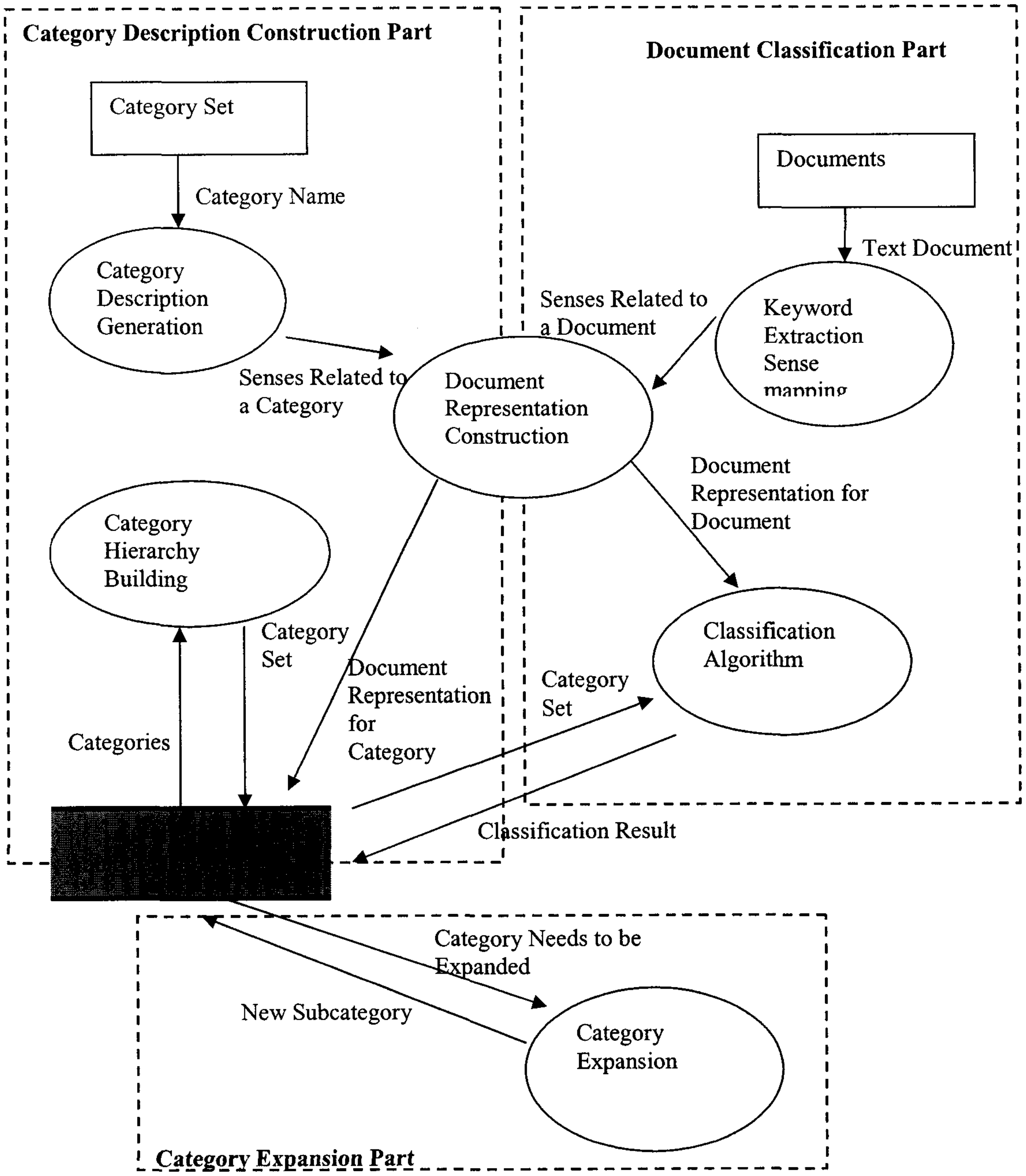

Category Description Construction Part
Category Set
Category Name
Category Description Generation
Senses Related to a Category
Category Hierarchy Building
Category Set
Categories
Document Representation for Category
Document Classification Part
Documents
Text Document
Keyword Extraction Sense mapping
Senses Related to a Document
Document Representation Construction
Document Representation for Document
Classification Algorithm
Category Set
Classification Result
Category Needs to be Expanded
New Subcategory
Category Expansion
Category Expansion Part

Figure 1.1 System Overview

CHAPTER 2

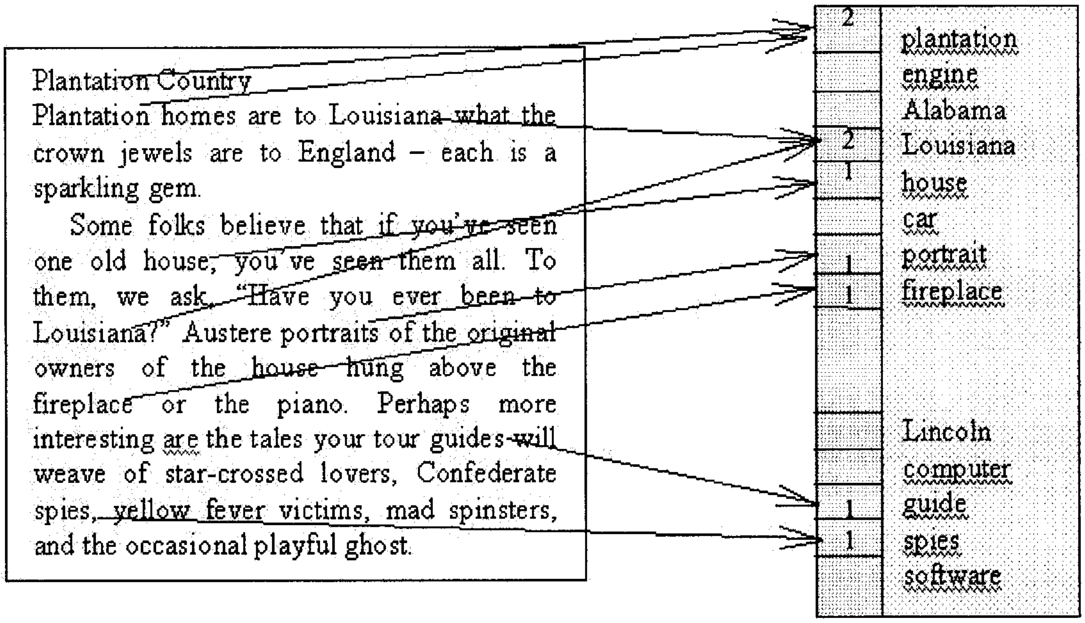
RELATED RESEARCH

The work on this dissertation is related to areas of text data processing, information retrieval, text clustering, and document classification. In this chapter, background information on these areas is provided, and some existing solutions from each area are presented.

2.1 Text Learning

Text learning is a machine-learning method on text data that also combines information retrieval techniques and is often used as a tool to extract the true content of text data. The product of any text learning process is a machine-readable form of a given document, which is called its document representation.

A common and widely used document representation in the information retrieval and text learning area is the bag-of-words text document representation, the idea of which is found in Koller and Sahami [1998] and Lang [1995]. One of the drawbacks of this document representation is that word order and text structure are ignored. Therefore, a great deal of the information from the original document is lost. The result is that the text is rendered incoherent to humans in order to make it coherent to a machine-learning algorithm. The process of obtaining this document representation is simple. Each word in the document is extracted, and the number of occurrences is counted. After all words and numbers of occurrences of each word are available, another step can be conducted, namely, calculating the probability for each word in the text document. This step is optional, depending on different requirements. Then, a data structure in a vector form is used to contain all these words, each of which has an associated number of occurrences (or probability) for this word. This vector is then regarded as the document representation for the text document. Figure 2.1 shows a sample from the bag-of-words document representation of a short text presented by Choi and Yao [2004].



Plantation Country

Plantation homes are to Louisiana ~~what~~ the crown jewels are to England – each is a sparkling gem.

Some folks believe that if you've ~~seen~~ one old house, you've ~~seen~~ them all. To them, we ask, "Have you ever ~~been to~~ Louisiana?" Austere portraits of the ~~original~~ owners of the ~~house~~ hung above the fireplace or the piano. Perhaps more interesting are the tales your tour guides ~~will~~ weave of star-crossed lovers, Confederate spies, yellow fever victims, mad spinsters, and the occasional playful ghost.

2	plantation
	engine
	Alabama
2	Louisiana
1	house
	car
1	portrait
1	fireplace
	Lincoln
	computer
1	guide
1	spies
	software

Figure 2.1 A Sample from the Bag-of-Words

Many experiments have been done to improve the performance of the text document representation. For example, Mladenic [1998] extended the bag-of-words representation to a bag-of-features representation. She defined the features of a text document as a word or a word sequence. Chan [1999] also suggested that using word sequences other than single words is a better choice. The goal of using word sequences as features is to preserve the information left out of the bag-of-words. This representation, which is also called “feature vector representation” in Chan [1999], uses a feature vector to capture the characteristics of the document by an “n-gram” feature selection, which extracts word sequences with *i* consecutive words from the entire document during the i-th run, and the range of *i* is from 1 to n. A 3-gram feature selection in the following sample text:

“Searching the World Wide Web”

will be done in three runs. The first run extracts five words: “searching,” “the,” “World,” “Wide,” and “Web.” The second run will extract two consecutive words such as “searching the,” “the World,” “World Wide,” “Wide Web.” The last run will extract three words: “searching the World,” “the World Wide” and “World Wide Web.” The experiments of Mladenic [1998] show that features with two or three words occur most often among all features of different lengths in the Yahoo documents of a 5-gram selection.

In text learning, if the document is represented by a vector of feature values, selecting the essential features and eliminating less useful features becomes a major issue. The usage of n-gram feature selection actually enriches the dimension of the feature vector even further. As seen in the example of the previous paragraph, a word sequence with five words provides 13 features after a 3-gram feature selection. The high number of features will inevitably increase the complexity and calculation needed so that the whole process may slow down dramatically. Thus, methods to reduce the number of features have been explored.

The most frequently used methods to reduce the number of features are “stopping” and “stemming.” The idea of “stopping” is to eliminate those common words that occur often and mean little, such as articles or prepositions. The idea of "stemming," on the other hand, is to use a language-specific algorithm to find the same semantic root of different words, as in the example "compute" and "computes," which are considered to be the same feature.

Other approaches used to reduce the number of features, such as those described in Yang and Pedersen [1997], do not depend on language itself. They use a feature scoring measure in order to select only the informative features. The feature scoring method is commonly used in selecting important features when text learning is performed. Yang and Pedersen further compare five measures of feature selection in text categorization on similar bag-of-words document representations. They point out that, even if we eliminate most of the features of the feature vector, the experimental results are similar to those using a large subset of the entire feature vector. In addition, by applying simple frequency of feature after "stopping," Yang and Pedersen achieve very good results in classification accuracy. Mladenic [1998] has tested eleven different kinds of measures in the Yahoo database and has confirmed these findings. By her experimental results, Mladenic also suggests that the Odds ratio, because of its capability of "favoring features characteristic for positive examples," outperforms other measures in scoring features.

2.3 Classification Algorithms

For text classification, the Term Frequency–Inverse Document Frequency (TFIDF) method is often used. TFIDF document representation represents each document as a vector in the space of words that are taken from training documents. The term frequency $TF(f_i ,Doc)$ of a word f_i in a document Doc is calculated by counting the number of occurrences of f_i . Let T be the total number of documents and $DF(f_i)$ be the number of documents having the word f_i, the inverse document frequency of a word f_i, denoted by $IDF(f_i)$, is calculated by $IDF(f_i) = Log\frac{T}{DF(f_i)}$. Then the document is represented by a vector with each item calculated as $V(i) = TF(f_i, Doc)IDF(f_i)$. Based on this document vector model, the similarity between vectors is calculated by the cosine of the angle between two vectors for the purpose of classification [Salton and Buckley 1988].

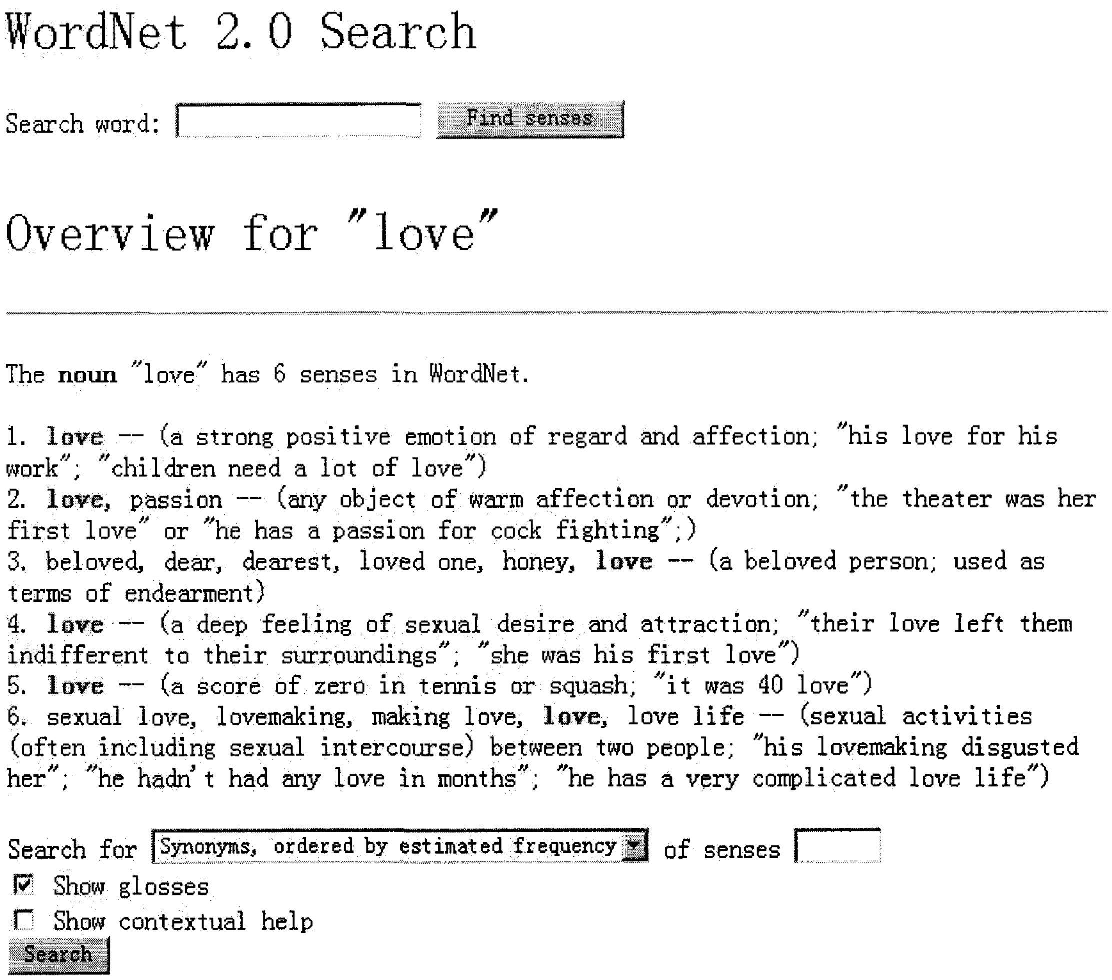
The TFIDF is extended by Joachimes [1997] who analyzed the TFIDF classifier in a probabilistic way based on the implicit assumption that the TFIDF classifier is as explicit as the naïve Bayes classifier. He proposed the PrTFIDF classifier by combining the probabilistic technique from statistic pattern recognition into the simple TFIDF classifier. The classifier optimizes the parameter selection in TFIDF and reduces the error rate by 40%, as reported in Joachimes [1997].

Support Vector Machines (SVMs) have shown good performance on different classification problems, and most recently, they have been used on text classification as seen in Joachims [1998], Dumais et al. [1998], Yang and Liu [1999], Sun et al. [2002], and Dewdney et al. [2001]. The classifier uses a structural risk minimization principle from computational learning theory, which can be found in Vapnik [1995] and Cortes and Vapnik [1995]. A text classification algorithm, which takes advantage of the hierarchical structure of categories, is reported in Choi and Peng [2004]. Other related classification methods can also be found in Choi and Yao [2004].

2.3 WordNet Database

In sense-based text classification, one of the important processes is mapping a word to corresponding senses. WordNet, developed by Princeton University, is an online lexicon database that can serve as a bridge from words to senses. The initial idea of WordNet, found in Miller [1990], Miller et al. [1990], and Beckwith [1990], was to change the classic way of searching dictionaries. Other than looking up a lexicon by the alphabet, WordNet provides a way to search dictionaries by the meaning of the lexicon. The latest version of the WordNet database found in WordNet Search [2.0] offers a simple interface that provides users the related senses to the input word. Figure 2.2 shows the search result after a user submits a search query for the word "love" to the database.

The search result shows the senses of the word with a clear definition and explanation for each sense. These senses are ordered so that the sense with the highest position will be the most commonly used one in a certain language.



WordNet 2.0 Search

Search word:

Overview for "love"

The **noun** "love" has 6 senses in WordNet.

1. **love** -- (a strong positive emotion of regard and affection; "his love for his work"; "children need a lot of love")
2. **love**, passion -- (any object of warm affection or devotion; "the theater was her first love" or "he has a passion for cock fighting";)
3. beloved, dear, dearest, loved one, honey, **love** -- (a beloved person; used as terms of endearment)
4. **love** -- (a deep feeling of sexual desire and attraction; "their love left them indifferent to their surroundings"; "she was his first love")
5. **love** -- (a score of zero in tennis or squash; "it was 40 love")
6. sexual love, lovemaking, making love, **love**, love life -- (sexual activities (often including sexual intercourse) between two people; "his lovemaking disgusted her"; "he hadn't had any love in months"; "he has a very complicated love life")

Search for Synonyms, ordered by estimated frequency of senses

☑ Show glosses

☐ Show contextual help

Figure 2.2 Senses for the Word "Love" in WordNet 2.0

The basic unit in WordNet is called a synonym set or *synset*. Each synset consists of a list of synonymous word forms. A word form in WordNet can be a single word or two or more words connected by underscores. According to the part of speech, all the synsets in the WordNet database are divided into several classes: nouns, verbs, adjectives, and adverbs. In each class, the synsets are organized by some semantic relations. Some of the relations used to construct the WordNet database are listed below:

Antonym: The "not-a" semantic relation, which refers to the synset with opposite meaning. This relation is symmetric. For example, *goodness* is the antonym of *badness*, and vice versa.

Hyponym / Hypernym: The "is-a" semantic relation or subset/superset relation. Hyponym is transitive and asymmetrical, comparable to the parent and child node in a tree structure. For example, *economics* is a hyponym of *social science,* but *social science* is a hypernym of *economics.*

Meronym / Holonym: The "has-a" relation. If the sentence "y has a part x" is meaningful, then x is the meronym of y and y is the holonym of x. For example, *table* has a *row*, then *row* is the meronym of *table* and *table* is the holonym of *row.*

The relationship that interests us here is the hypernym-hyponym relation between nouns. One synset is a hypernym of another if it covers a more general meaning. For example, *science* is a hypernym of *natural science* and *social science,* since it represents a more general concept. Based on this relation, all the noun synsets form a tree-like structure. An example of a small section of the WordNet database with respect to the hyponym-hypernym order on nouns is shown in Figure 2.3.

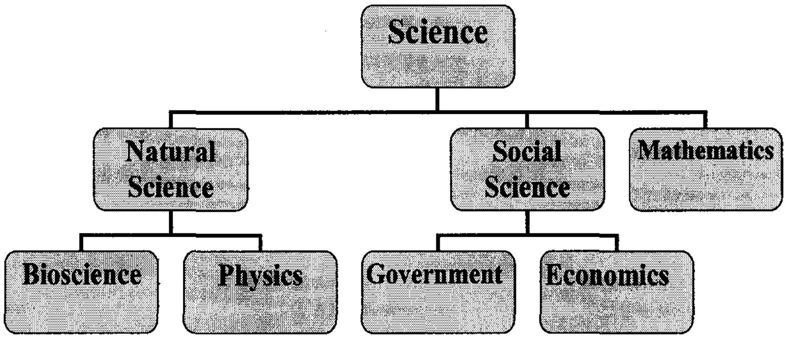


Figure 2.3 A WordNet Synset Tree Example

As WordNet provides a lexical database that maps words into synsets semantically, it is widely used for sense-based projects. In the text classification area, WordNet is also used to construct the document representation. With the help of WordNet, Rodriguez et al. [1997] used the synonym and showed an improvement in classification accuracy on a collection of documents that appeared on Reuter's newswire in 1987 [Reuters-21578]. Scott and Matwin [1998] used both synonym and hypernym to develop a *hypernym density representation*. Their experiments in three different testing databases achieved a marginal improvement in accuracy.

2.4 Hierarchical Structure Information Propagation

Tree structures are used extensively nowadays to depict all kinds of taxonomic information, such as the Yahoo category structure and the file management system in Microsoft Windows [Microsoft]. In this kind of application on a tree structure, the child node is a subdivision of the parent node, which usually represents information that is more general. Then, the information in the child node should also be considered as part of the information of the parent node because of the parent-child relationship. For example, if the parent node is "fruit," then one of the children nodes can be "apple" because "apple" is a subdivision of "fruit." Then the information "green apple" existing in the "apple" node should also be considered as information of the "fruit" node since "green apple" is a "fruit."

If the presence of the information in a child node can be represented by an *original weight*, then one way to present the existence of this information in the parent node is to assign a scale factor to the original weight and propagate it to the parent node. The scale factor reflects the parent-child relationship. If an original weight is in each tree node of a tree structure, then the propagated weight in the tree structure, which contains the hierarchical information, can be estimated by propagating all the original weights following the parent-child relationships from the leaf nodes to the root. The value is called the *propagated weight* of the tree and is assigned to the root of the tree structure.

Incorporating the structural information will improve the performance of text classification as reported in Peng and Choi [2002], Mladenic [1998], and Koller and Sahami [1998]. One of the solutions of capturing the category hierarchy information can be found in Mladenic [1998]. She analyzes the Yahoo category structure and presents a formula to assign scale factors for each category in the category hierarchy based on the number of URLs in each category and the position of the category in the hierarchy. Then, a recursive function is used to calculate the propagated weight of each keyword or keyword sequence in each category. In this dissertation, we modify the algorithm to capture the hierarchical information of a tree structure and apply it to constructing the new semantic hierarchy representation. Details on the modified algorithm and applications can be found in Chapter 3 and Chapter 4.

2.5 Sense Disambiguation

WordNet can be used to map keywords to senses, but, unfortunately, many English words do not have a one-to-one mapping between spelling and meaning [Wnstats]. The latest WordNet version 2.0 has 152,059 unique words and word sequences. The number of words with more than one sense is 26,275 [Wnstats]. The problem of automatically detecting the correct sense for a word form in a context is called “word sense disambiguation” (WSD) as found in Yarowsky [1992], Agirre and Rigau [1996], and Ganesh et al. [2004].

Searching for a good solution for word sense disambiguation seems to be a very difficult task. Bar-Hillel [1960] even declares that the solution to determining the correct sense of the word *pen* in the sentence “*The box is in the pen*” does not exist. Ide and Veronis [1998] described the problem as *AI-complete,* which means that a problem can be solved only after resolving all the difficult problems in artificial intelligence (AI), such as the representation of common sense and encyclopedic knowledge.

However, in the past two decades, with developments in several related areas such as natural language processing (NLP), knowledge representation, text learning, and information retrieval, the likelihood of finding a solution for the automatic word sense disambiguation has become more probable [Ide and Veronis 1998]. In the past ten years, large amounts of machine-readable text have been processed and become available because the tremendous improvement in computer calculating power. With the help of statistical methods developed in these ten years, more and more information about regularities in this machine-readable text data is recognized. Therefore, attempts to disambiguate word sense automatically have increased [Ide and Veronis 1998].

In general, the problem of word sense disambiguation can be classified into four different strategies: AI-based, knowledge-based, corpus-based, and hybrid strategy.

The AI-based strategy became popular in the 1960's [Quillian 1961, 1962]. This strategy takes natural language understanding as the first step and uses a large AI system and many testing samples to model the syntax and semantics of human languages. Inevitably, the knowledge sources required for AI-based systems should be done by manpower [Waltz and Pollack 1985]. Therefore, most of the AI-based systems do not have a satisfying disambiguation power, and the experiments are always limited to a small context [Waltz and Pollack 1985]. This limitation makes the application of AI-based sense disambiguation in real text data very difficult.

In the knowledge-based approach, the disambiguation task is carried out by using a knowledge base, or an explicit lexicon [Klavans et al. 1990]. The lexicon may be a thesaurus, machine-readable dictionary, or even a handcrafted database. Many algorithms have been implemented using existing lexical knowledge sources such as WordNet, [Agirre and Rigau 1996, Resnik 1995], LDOCE [Cowis et al. 1992, Gutherie et al. 1991], and Roget's International Thesaurus [Yarowsky 1992]. When compared to the AI-based strategies, this strategy automatically extracts information directly from those lexical knowledge sources, avoiding the complex process of semantic rule analysis. This advantage makes the knowledge-based approach one of the most popular approaches, as seen in Ide and Veronis [1998], to word sense disambiguation.

The corpus-based approach obtains the sense information by applying a training technique on some text data corpus, instead of getting it from the existing knowledge base. The training corpus can be either a *disambiguated* or a *raw* corpus. In a disambiguated corpus, each lexical item with several meanings is marked. A raw corpus does not have this marked lexical item. This approach requires more computation resources than the knowledge-based approach, as seen in Levow [1997], because of the training process.

The hybrid approach is a combination of knowledge-based and corpus-based approaches. Luk's system [Luk 95] is a good example of this strategy. He collects the textual definitions of senses from a machine-readable dictionary (LDOCE) to identify relations between senses. He then calculates mutual information scores between these related senses by training in a corpus. The score information is an indicator to discover the most useful sense. As a result, the system uses the information in lexical resources as a way of reducing the amount of text needed in the training corpus.

2.6 Clustering Techniques

Clustering is the process of segmenting a set of objects into different subsets whose members share the same character. A *cluster* is therefore a small set of objects which are "similar" to all other objects within the cluster and are "dissimilar" to the objects belonging to other clusters. Clustering techniques are widely used in the text analysis domain to group similar text documents. The text clustering process includes two steps: the first step is defining a measure to capture the relation between documents, which we further describe in Section 2.6.1. The second step is applying a different clustering algorithm based on a relation matrix of the chosen measure, which is presented in Section 2.6.2.

2.6.1 Similarity Measures

An important component of a clustering task is the relation measure between data points. For a clustering task on text documents, all the text documents are turned into a vector form. According to this document representation, distance measure and similarity measure are often used to capture the relation of two text documents.

For higher dimensional data, a popular measure is the Minkowski metric, as found in Baez and Dolan [1995]. Let p be an integer, the p-norm between two vectors $x_{i,}$ x_j is denoted D_p, which can be calculated by Formula 2.1.

$$D_p(x_{i,}x_j) = \left(\sum_{k=1}^{n} \left\| x_{ik} - x_{jk} \right\|^p \right)^{1/p}$$

Formula 2.1 The Minkowski Metric Formula

In Formula 2.1, $x_{ik,}$ x_{jk} are the k-th element of two vectors, n is the dimensionality of the data, and the "|| ||" in the formula is the absolute value operation. As special cases, *Euclidean* distance is taken where p=2, while *Manhattan* metric has p=1. However, no general theoretical guidelines have been developed for selecting a measure for any given application [Baez and Dolan 1995].

Another popular measure of similarity for text clustering is the cosine of the angle between two vectors $x_{i,}$ x_j. The cosine measure of two vectors is given by Formula 2.2.

$$\boxed{Sim(x_i, x_j) = \frac{\sum_{l=1}^{n}(x_{il}x_{jl})}{\sqrt{\sum_{l=1}^{n} x_{il}^2}\sqrt{\sum_{l=0}^{n} x_{jl}^2}}}$$

Formula 2.2 The Vector Similarity Formula

This cosine similarity does not depend on the lengths of the two vectors. In addition, because of this property, samples can be normalized to the unit sphere for more efficient processing, as pointed out in Dhillon and Modha [2001].

2.6.2 Clustering Algorithms

Clustering techniques can be broadly categorized into two classifications: non-hierarchical methods and hierarchical methods [Everitt et al. 2001; Jain et al. 1999]. The major difference between them is whether they produce flat partitions or a hierarchy of clusters. The k-means method is the most popular non-hierarchical clustering algorithm, as it has O(n) time complexity in terms of the number of data points [Steinbach et al. 2000, Dhillon et al. 2001]. The k-means method assigns data points to clusters in such a way that the mean square distance of points to the centroid of an assigned cluster is minimized. The problem with the k-means method is that it is very sensitive to outliers as pointed out in Choi and Yao [2004]. The medoid-based method, on the other hand, solves this problem by trying to find the most center points to represent clusters [Ng and Han

1994]. But these methods have O(n^2) complexity, as pointed out by Berkhin [2002]. These two non-hierarchical clustering methods share some problems that need to be considered. First, the results of these two methods are sensitive to the number of resulting clusters and initial seeds. These two methods do not work well when clusters have either a large variation in size or arbitrary shapes.

In hierarchical clustering, the data are not partitioned into a particular cluster in a single step. Instead, this clustering method may run from a single cluster containing all documents followed by a top-down divisive method or by using n clusters, each containing a single object followed by an agglomerative method. Hierarchical agglomerative clustering (HAC) algorithms are more popular than the divisive ones. The primary difference in HAC is the way that they compute the similarity (or distance) between clusters [Jain et al. 1999; Strehl 2002; Karypis et al. 1999].

2.6.3 Clustering Quality Measures

In the information retrieval community, three standard measures are widely used to judge the quality of a cluster: precision, recall, and F1-value. Precision is a measure of the purity of the cluster, and recall is a measure of the completeness of the cluster retrieval. To evaluate the clustering performance, a group of *n* documents, within which *m* documents are from a certain defined category c, should be given. If the result of the clustering is a cluster of *k* documents, in which *l* documents are from category c, then the precision for this cluster is $P = \frac{l}{k}$, and the recall is $R = \frac{l}{m}$. The F1-value combines precision and recall with equal weights into a single number, which is defined as $F_1 = \frac{2PR}{P+R}$. A clustering method that has a good performance will have a high value of these three measures.

CHAPTER 3

CATEGORY DESCRIPTION CONSTRUCTION

3.1 Introduction

For a classification system, a category hierarchy should be given as the first step. Most of the category hierarchies are in a human-readable form. For an automatic classification system, the information of the category hierarchy should be turned into a machine-readable format. In order to extract the information of the category hierarchy, three issues need to be considered: sources for category content information, category representation, and category structure information.

The first issue determines the sources for generating content information for each category. In many existing classification systems, such as Koller and Sahami [1998] and Mladenic [1998], a set of example documents is used to generate the category information. This method has two major problems. If a large number of examples is used, because of the variety of the examples, the category description might contain a lot of unrelated information. This results in a problem called *information pollution*. On the other extreme, if only a small number of examples are used, the category description might not have enough words or senses to cover the content of the category. This results in another problem called *insufficient information*. For our sense-based classification system, we use the synsets containing the category name as the main source to describe the category information. Then, the category name synsets are enriched by semantic related synsets from WordNet. As a result, a category is turned into a group of synsets, each of which is associated with a weight. Details on the process of describing a category are provided in Section 3.2.

The second issue is choosing a machine-readable representation for the information of each category. If senses are used to represent a document, the connection between senses plays a key role in capturing the ideas in the document. Recent research [Kehagias et al. 2001] shows that simply changing the keywords to senses (bag-of-sense) without considering the relation between senses does not have a significant improvement over the traditional keyword-based classification method. In some special cases, the sense-based classification method performs worse than the keyword-based classification method [Kehagias et al. 2001]. Differing from existing sense-based approaches, we present an algorithm to construct a new sense-based representation called semantic hierarchy representation in Section 3.3. This representation uses a formula to assign different scale factors for each synset in the WordNet synset structure and captures the synset hierarchy information between synsets by propagating the weight of each synset to its hypernym synsets according to these scale factors.

The third issue is capturing the category hierarchy information. Most current classification systems ignore the hierarchy information. As found in Peng and Choi [2002], the information of the category/subcategory relations in the category hierarchy will contribute to a better classification result. In order to capture the category hierarchy information, the scale factor assigning and weight propagation formulas in Section 3.3

are applied again on the category hierarchy. This process is described in detail in Section 3.4

When a predefined category hierarchy is given, we use the process in Figure 3.1 to generate the category description for our classification system.

```
For each category in the predefined category hierarchy
        Generate the synsets for category from category name
        Construct the sense-based representation for the category
 Capture the hierarchical information by propagation
```

Figure 3.1 Pseudo Code for Category Description Construction

3.2 Generate the Synsets for Category from Category Name

We use a group of synsets to describe a category, and these synsets are extracted from three resources: the synsets containing the name of the category, the meronym synsets of the category name synsets, and synsets containing keywords from the explanations of category name synsets and their meronyms. Among these synsets, the category name synsets are assigned a weight of two because of the importance of these senses. The weights of all other synsets are assigned as one. After the initial assigning of weight for each synset, we then give a percentage to each synset based on its weight. The algorithm is described in the pseudo code in Figure 3.2.

```
Given category name, retrieve synsets containing the category name from WordNet.
Increase the weight of the synsets corresponding to the category name by two
    For each of the synsets
        Retrieve the keywords from the explanation of the category name synset
        For each of the keywords
            Retrieve the synsets containing the keyword
            Increase the corresponding weight by one
        Retrieve the meronyms of the synset
        For each of the meronyms
            Increase the weight of synset of the meronym by one
            Retrieve the keywords from the explanation of the meronym
            For each of the keywords
                Retrieve the synsets containing the keyword
                Increase the weight of the corresponding synset by one
Calculate the probability of each synset based on its weight
```

Figure 3.2 Pseudo Code for Generating Related Senses from Category Name

In the last step of the pseudo code, the probability (p_i) is assigned to the i-th sense according to the weight of each synset (w_i) and the total number of synsets (n) by using $p_i = \frac{w_i}{\sum_{i=1}^{n} w_i}$. A category now is represented by a weight distribution of noun synset hierarchy in WordNet.

3.3 Construct the Representation for a Category

In this section, we develop a semantic hierarchy representation, which makes use of the noun synset hierarchy provided in WordNet, for each category. In the last section, a category is presented by a weight distribution of noun synsets in WordNet. In the WordNet database, these synsets are organized into a tree structure by the hypernym-hyponym relation. As mentioned in Section 2.4, if a distribution in a tree structure is represented by an original weight in each tree node, then the propagated weight can be calculated to capture the hierarchical information. Based on the research by Mladenic [1998] on Yahoo category structure, we present the modified propagation method to capture the information of noun synset hierarchy in the WordNet database by using a weight propagation formula (Formula 3.1) and a scale factor assigning formula (Formula 3.2). After the propagation process, as described by pseudo code in Figure 3.3, the category is turned into the semantic hierarchy representation, which is a propagated weight distribution of noun synsets in WordNet.

```
For each synset from the category information
    Calculate the propagated weight for the synset according to Formula 3.1
```

Figure 3.3 Pseudo Code for Representation Construction for Category

We consider the WordNet noun synset hierarchy as a tree T by taking each synset as a tree node. A subtree T_N, whose root node is tree node N, has k children nodes labeled from N_1 to N_k. As a special case, when k=0, T_N can be considered as a *leaf node*. T_N has k *direct subtrees*, whose root nodes are N_1 to N_k correspondingly. $Sub_N(N_i)$ is the direct subtree rooted at child node N_i. Then, the probability associated with synset N after the category description generation mentioned in Section 3.2 is treated as the original weight

W(N). The propagated weight *W'(T_N)* for tree T_N is calculated using Formula 3.1(based on Mladenic [1998]).

$$W'(T_N) = W(N)\alpha(N,T_N) + \sum_{i=1}^{k} W'(Sub_N(N_i))\beta(Sub_N(N_i),T_N)$$

Formula 3.1 Formula for Propagated Weight Calculation on WordNet

Formula 3.1 is used recursively starting from the root N and stops in the leaf nodes. In Formula 3.1, the propagated weight of the tree T_N is composed by the original weight of the root node multiplied by a scale factor $\alpha(N, T_N)$ and the propagated weight of each direct subtree multiplied by the corresponding scale factor $\beta(Sub_N(N_i),T_N)$. These two types of scale factors are calculated using the size of node N (*Size(N)*) and the size of each direct subtree (*Size(Sub_N(N_i))*) following Formula 3.2. The size of node N is defined as the number of synonyms within the synset corresponding to node N. Correspondingly, Size($Sub_N(N_i)$) is the number of synonyms within all the synsets in the subtree $Sub_N(N_i)$.

$$\alpha(N,T_N) = \frac{\ln(1 + Size(N))}{\ln(1 + Size(N)) + \sum_{i=1}^{k} \ln(1 + Size(Sub_N(N_i)))}$$

$$\beta(Sub_N(N_i),T_N) = \frac{\ln(1 + Size(Sub_N(N_i)))}{\ln(1 + Size(N)) + \sum_{i=1}^{k} \ln(1 + Size(Sub_N(N_i)))}$$

Formula 3.2 Formulas for Assigning Scale Factors

As a simple demonstration of the propagated weight calculation process, a simplified synset hierarchy in WordNet with the original weight and the size of each synset is presented in Figure 3.4. The propagated weight in each node is calculated using Formula 3.1 and Formula 3.2. Because the calculations on leaf nodes are simple and will be used for its upper level, we start the calculation from the bottom. The calculation process and results are listed in Figure 3.5

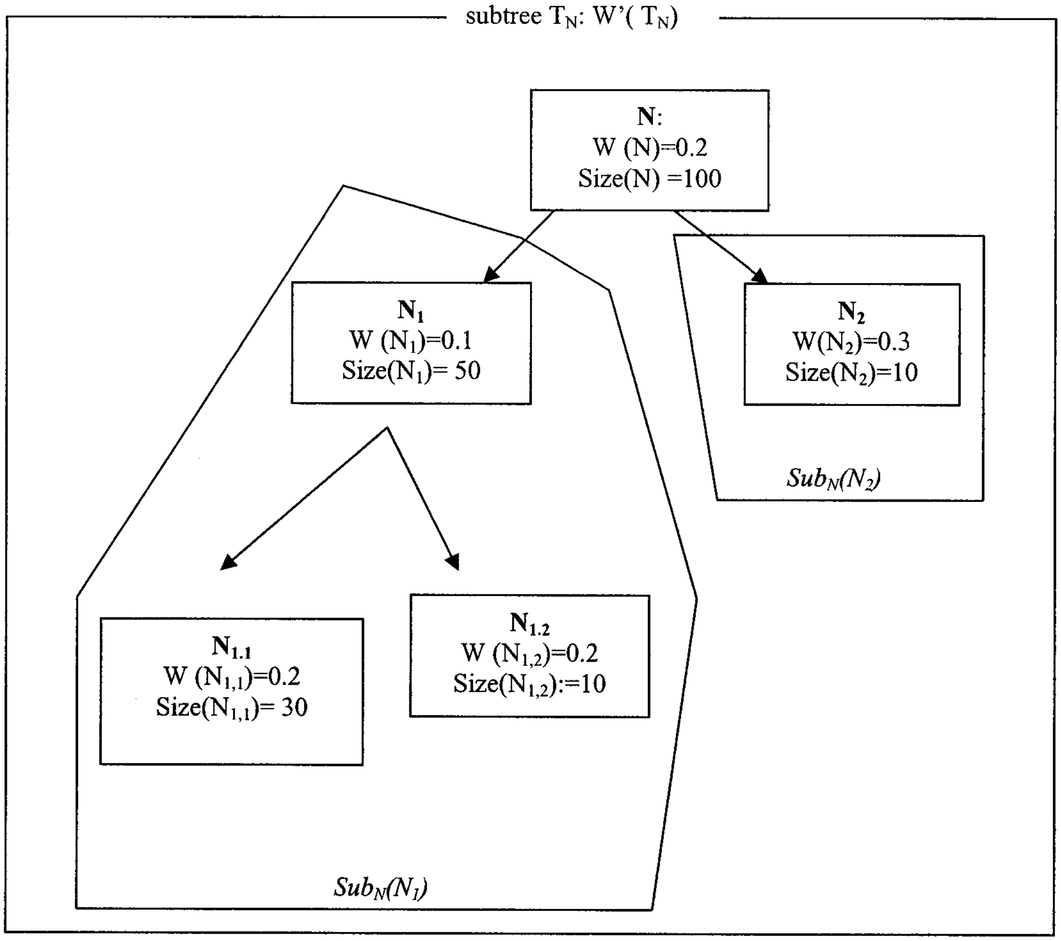


Figure 3.4 A Simplified Synset Tree in WordNet

$$
\begin{aligned}
&Leaf \quad nodes: \\
&W'(Sub_{N_1}(N_{1.1})) = W(N_{1.1}) = 0.2 \\
&W'(Sub_{N_1}(N_{1.2})) = W(N_{1.2}) = 0.2 \\
&W'(Sub_N(SN_2)) = W(N_2) = 0.3
\end{aligned}
$$

$$
\begin{aligned}
&Node \quad SN_1: \\
&\alpha(N_1, T_{N_1}) = \frac{\ln(1+50)}{\ln(1+50)+\ln(1+30)+\ln(1+10)} = 0.4027 \\
&\beta(Sub_{N_1}(N_{1.2}), T_{N_1}) = \frac{\ln(1+10)}{\ln(1+50)+\ln(1+30)+\ln(1+10)} = 0.2456 \\
&W'(Sub_N(N_1)) = 0.4027W(N_1) + 0.3517W(N_{1.1}) + 0.2456W(N_{1.2}) \\
&= 0.1597
\end{aligned}
$$

$$
\begin{aligned}
&Node \quad N: \\
&\alpha(N, T_N) = \frac{\ln(1+120)}{\ln(1+120)+\ln(1+90)+\ln(1+10)} = 0.4097 \\
&\beta(Sub_N(N_1), T_N) = \frac{\ln(1+90)}{\ln(1+120)+\ln(1+90)+\ln(1+10)} = 0.3854 \\
&\beta(Sub_N(N_2), T_N) = \frac{\ln(1+10)}{\ln(1+120)+\ln(1+90)+\ln(1+10)} = 0.2049 \\
&W'(T_N) = 0.4097W(N) + 0.3854W'(Sub_N(N_1)) + 0.2049W'(Sub_N(N_2)) \\
&= 0.2050
\end{aligned}
$$

Figure 3.5 An Example of the Propagated Weight Calculation

3.4 Capturing Category Hierarchy Information

If the category hierarchy is a tree structure, to capture the hierarchy information, the category description of a category should be propagated to its parent category. After the semantic hierarchy construction, a category is represented by a distribution of weight in the WordNet synset hierarchy. As each category has the same representation, the propagation process to capture the category hierarchy information can be accomplished by propagating the weight of each synset in a category representation to the corresponding synset in the parent category following Formula 3.3. The propagation process is listed as pseudo code in Figure 3.6.

```
For each category in the category hierarchy structure
    For each synset from the semantic hierarchy representation of the category
        Calculate the propagated weight for the synset according to Formula 3.3
```

Figure 3.6 Pseudo Code for Capturing Category Hierarchy Information

We consider the category hierarchy as a tree T_C rooted at C by taking each category as a tree node. C has k children nodes labeled from C_1 to C_k. T_C has k direct subtrees labeled $Sub_C(C_1)$ to $Sub_C(C_k)$, whose root nodes are C_1 to C_k correspondingly. The weight associated with a synset N in category C after the semantic hiearchy construction mentioned in Section 3.3 is treated as the original weight $W_N(C)$. Then, propagated weight $W_N'(T_C)$ for this synset in the semantic hiearchy in category C with category hierarchy information is calculated using Formula 3.3.

$$W_N'(T_C) = W_N(C)\alpha(C,T_C) + \sum_{i=1}^{k} W_N'(Sub_C(C_i))\beta(Sub_C(C_i),T_C)$$

Formula 3.3 Formula for Propagation on Category Hierarchy

Formula 3.3 is used recursively starting from the root C and stopping in the leaf nodes. In Formula 3.3, the scale factor $\alpha(C, T_C)$ for tree node C and the scale factor $\beta(Sub_C(C_i), T_C)$ for a direct subtree are calculated by the size of node C ($Size(C)$) and the size of each direct subtree ($Size(Sub_C(C_i))$) following Formula 3.4. The size of node C can be defined according to different classification tasks, such as existing documents or URLs within category C. Correspondingly, $Size(Sub_C(C_i))$ is the number of existing documents or URLs within all the categories in the direct subtree $Sub_C(C_i)$ of C.

$$\alpha(C, T_C) = \frac{\ln(1 + Size(C))}{\ln(1 + Size(C)) + \sum_{i=1}^{k} \ln(1 + Size(Sub_C(C_i)))}$$

$$\beta(Sub_C(C_i), T_C) = \frac{\ln(1 + Size(Sub_C(C_i)))}{\ln(1 + Size(C)) + \sum_{i=1}^{k} \ln(1 + Size(Sub_C(C_i)))}$$

Formula 3.4 Formulas for Assigning Scale Factors in Category Hierarchy

The propagation process using Formula 3.3 and Formula 3.4 are applied on each synset in the semantic hierarchy representation of category *C* to capture the category hierarchy information of T_C. After all categories are processed in a similar way, each category will have a new weight distribution on the WordNet synset hierarchy. This weight distribution contains the category hierarchy information as well as the WordNet synset hierarchy information. We use this semantic hierarchy representation for the category hierarchy. This way, the category description hierarchy is established and ready for classification purposes. Figure 3.7 gives an abstract view of the predefined category description in a tree hierarchy after the category description construction described in this section. Each category is represented by a semantic hierarchy according to the WordNet structure, and the category hierarchy is another tree structure. The information of these two hierarchical structures is captured by applying the propagation function on the weight of each synset in a category.

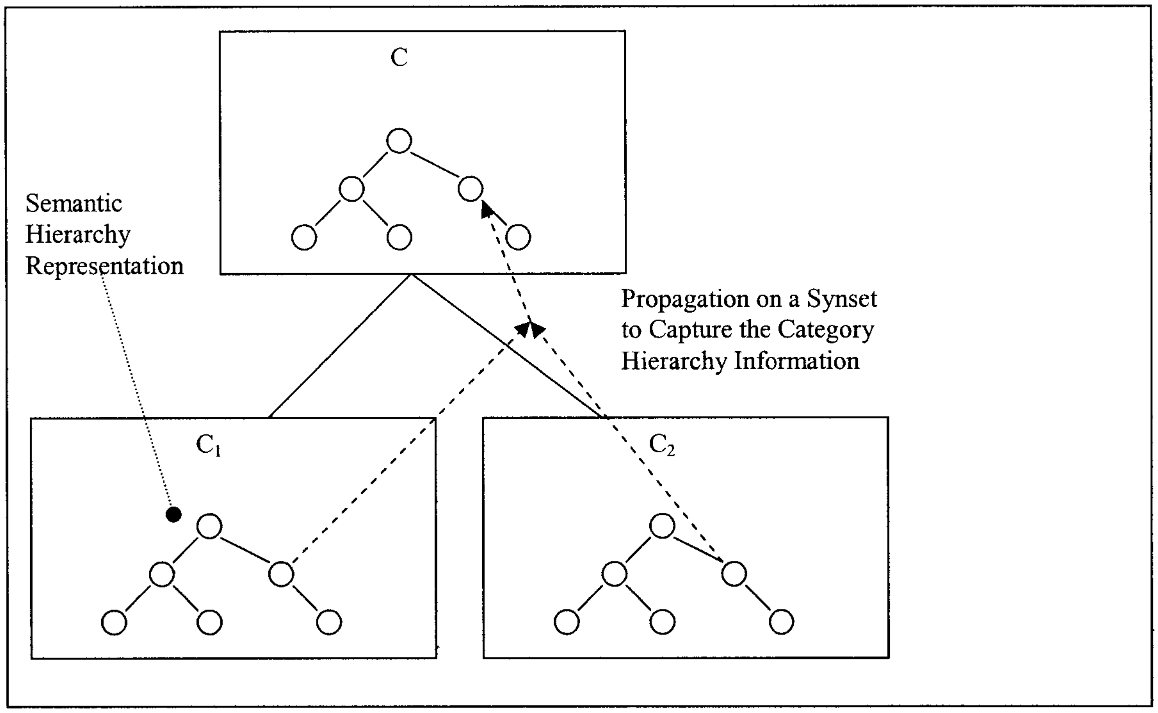

C
Semantic Hierarchy Representation
Propagation on a Synset to Capture the Category Hierarchy Information
C1
C2

Figure 3.7 The Predefined Category Hierarchy

CHAPTER 4

SEMANTIC HIERARCHY CLASSIFICATION

4.1 Introduction

In Chapter 3, the information of the category and the category hierarchy is represented by semantic hierarchy representation. In this chapter, we focus on classifying a document to the predefined category hierarchy. To classify a given document, the first step is extracting the keywords from the document. We present a new keywords extraction method in Section 4.2 for this purpose. In Section 4.3, a WordNet-based sense disambiguation algorithm is introduced to select the correct synset for each keyword, and the probability is then calculated. Then, in Section 4.4, based on these probabilities for the disambiguated synsets, the document is turned into the semantic hierarchy representation. We present the classification algorithm in Section 4.5. The similarity of the document and each category is calculated. Then, we select the category with maximum similarity as the category for the document. The whole process of the classification algorithm is summarized in the pseudo code list in Figure 4.1.

```
For each document
    Extract keywords from the document
    Sense disambiguation based on WordNet
    Construct the Semantic hierarchy Representation on disambiguated senses
    Classify the document based on the given category hierarchy
```

Figure 4.1 Pseudo Code for Classification Algorithm

4.2 Extract Keywords from a Document

In this section, we present a new approach to extract appropriate keywords from documents. Our approach incorporates two additional steps that are ignored by all known keyword extraction methods. As in Figure 4.3, the first step is segmenting the whole document into smaller text units. A text unit can be a sentence or part of a sentence. Details are provided in Section 4.2.1. The second step is to assign different weights to the text unit. We realize that not all the text units are equally important. The content, HTML tags, and URL of a text document might help in deciding a correct importance rate on different text units and enabling the acquisition of a better document representation. In Section 4.2.2, different types of text units are discussed, and weights are assigned accordingly. The remaining two steps are to apply n-gram selection and stop words removal on the text units. These steps are described in Section 4.2.3. The keywords extraction process can be summarized by the pseudo code in Figure 4.3

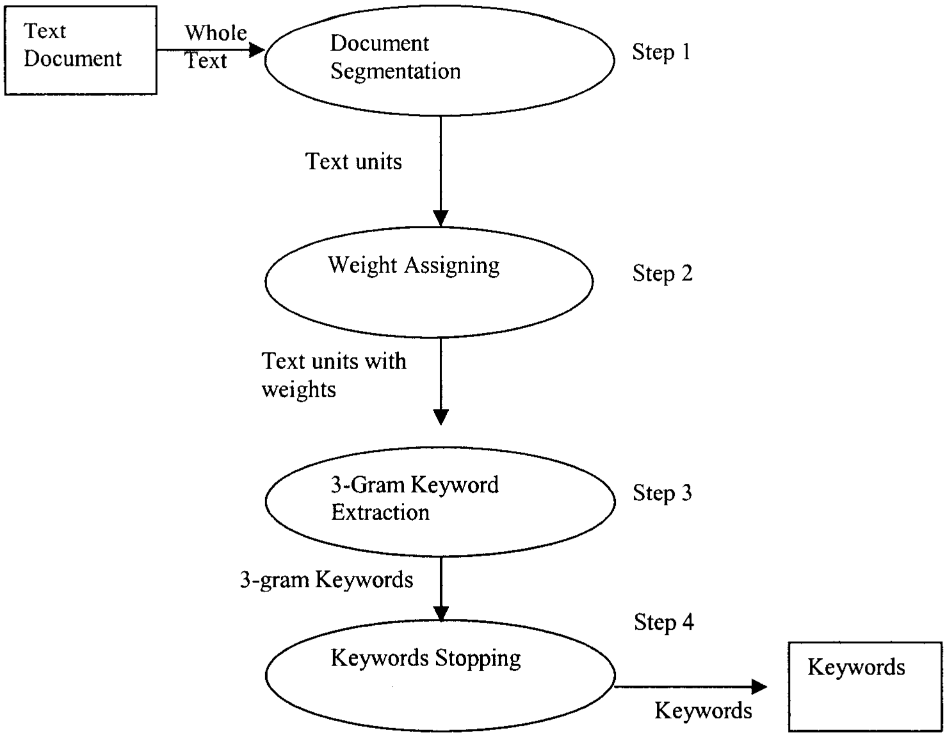


Figure 4.2 Keywords Extraction Process

```
Segment a document to text unit
For each text unit in the document
    Assign weight to the text unit
    Apply the n-gram selection on the text unit
    Remove stop words
```

Figure 4.3 Pseudo Code for Retrieving Keywords from a Document

4.2.1 Segment a Document

In order to segment a document to smaller units, we analyze the entire document and find all the delimiters such as ‘,’ ‘.’ ‘!’ ‘?’ ‘”’ ‘:’ ‘;’ and other delimiting symbols except spaces. Then, the text between two delimiters is considered as a text unit. In this way, a document is turned into a number of smaller text units.

Segmenting the document will reduce the number of word sequences when applying the n-gram selection because the words separated by a delimiter will not be combined to form a word sequence. The second advantage is that segmenting the document will reduce unrelated words. For example, a sentence fragment like "...spiders are known *world wide*. The *web* of different kinds of spiders..." might contribute a word sequence "World Wide Web" after removing stop word "the" and applying 3-gram feature selection technique on the entire sentence. The problem is obvious that the topic of the document is about insects and has nothing to do with the World Wide Web.

4.2.2 Assign Weights on Text Units

After the document is segmented to smaller text units, we need to recognize different levels of importance between text units. A text unit is considered to be more important in a text document, meaning that it can describe the main topic of the document better. For example, the text units within the title line are usually considered more important than those text units in the normal body text. The common way to specify different importance of text units is by assigning higher weights to text units that are more important. Several different sources of information within the text document itself will help with assigning weights to different text units in a text document. To be more specific, the information from document context, HTML tags, and URLs are three major information sources that can be used to determine the weights for text units. We analyze the text document and assign different weights listed in Table 4.1 according to different types of text units.

Table 4.1 Different Weights of Text Units

Text unit type	Weight assigned to text unit
Normal body text	1
First and last paragraph of a document	2
Beginning sentence of a paragraph	2
Sentences with bonus words and indicator phrases as pointed out in Paice [1990]	3
<title>	4
<H1>	2
<H2>	1
<EM>	2
<Strong>	3
<Meta >	3
URL	4

In Table 4.1, the weight of the normal body text is set to one. Then, based on the research on text documents found in Paice [1990], we assign higher weights to three types of text units that are important in a text document. These three types of text units are sentences using bonus words such as "greatest," significant, or indicator phrases such as "the main aim of," "the purpose of," sentences appearing at the beginning, or the last paragraph of the document and sentences appearing at the beginning of each paragraph.

The HTML tags also provide many clues about the importance of the text unit by using different tags. Some tags are rather straightforward, such as <title> for title, <h1>~<h6> to different kinds of headings, and <em> for emphasis. These HTML tags can be used as importance indicator tags. The <meta> tag, on the other hand, is "invisible" to users who read the web page in the browser, but is actually a good source to extract important text units. Pierre [2000] has sampled nearly 20,000 web pages and pointed out that about one third of the pages contain informative meta tags with forms like *<META NAME="description" CONTENT=" ......">* or *<META NAME="keywords"*

CONTENT="……">. When some text is quoted by those importance indicator tags, or if it is within the meta tag specified by the forms mentioned above, it can be considered as an important part of the document and given a higher weight in Table 4.1.

We assign a higher weight to the text units within the URL, which is an abbreviation of *Uniform Resource Locator*s as defined by [RFC 1738]. A URL is the simple and highly condensed way to provide information. When a web master names a web site, he or she will tend to develop a name related to the content of the whole web site. The path and file name part of a URL contains no special syntax. Since the path structure and the naming of the path where the file resides are unlimited, people will use the words that remind them of the file content to name the files and folders in order to find the document easier. This fact provides a good base to identify the important words that relate to the document. Once a keyword appears in a URL, the possibility of classifying that URL to the related category is higher than those keywords shown in HTML files. In order to employ this fact, the text unit in the URL are assigned a weight of four as seen in Table 4.1.

4.2.3 Apply the n-Gram Selection and Remove Stop Words

After segmenting a document to smaller units and weighting it accordingly, we apply the 3-gram keywords selection and filter out the stop words on the text units. The 3-gram keywords selection extracts word sequences with *i* consecutive words from the entire text unit during the i-th run, and the range of *i* is from 1 to 3. With the help of the stop word list taken from lextek.com [lextek], we remove the stop words from the keyword list as the last step of the keyword extraction process.

4.3 Sense Disambiguation Based on WordNet

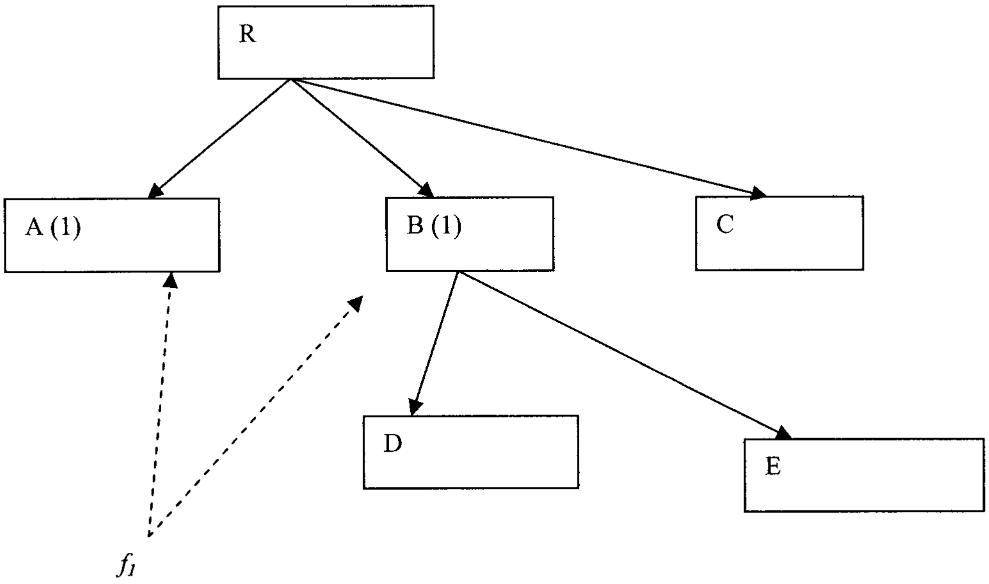
After keywords are extracted from a document that needs to be classified, the synsets containing those keywords will be retrieved after mapping each of the keywords to WordNet. As one word may have several meanings, one word may be mapped into several synsets in the WordNet database. In this case, we need to determine which meaning is being used, which is a problem with sense disambiguation. Because a sophisticated solution for sense disambiguation is usually expansive, we present a naïve approach based on WordNet.

Our sense disambiguation method consists of four passes. In the first pass, a keyword *f* is simply mapped into synsets that contain *f*, and the weights of these synsets will be increased by one unit. In the second pass, for each of the synsets containing *f*, we add one unit to the weights of the hyponym synsets and hypernym synsets. The third pass is to repeat pass one and pass two for each of the keywords in the keyword list needing to be disambiguated. This pass will cause overlapping, which is represented by the value of weight for each synset. The last pass is to check all the synsets associated with *f* and select the synset containing *f* with the highest weight as the most relevant one. If all the synsets of a word have equal weight, with the help of the sense ordering in WordNet, we select synset number one, which represents the most often-used synset of the keyword. The pseudo code is listed in Figure 4.4

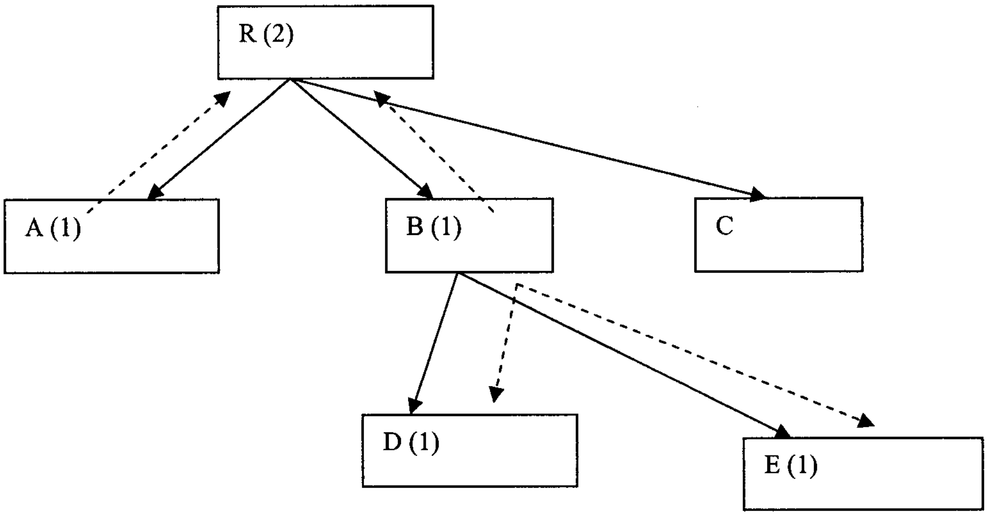
```
For each keyword in the keyword list
    Map this keyword to synsets by WordNet
    For each synset containing this keyword
        Increase the weight of this synset by one
        Retrieve the hypernym and hypernym synsets of current synset
        For each synset in the hypernym and hypernym synsets
            Increase the weight of this synset by one
For each keyword in the keyword list
    Select the synset with maximum weight as the disambiguated synset.
For each disambiguated synset
    Calculate the probability for this synset by occurrences
```

Figure 4.4 Pseudo Code for Sense Disambiguation

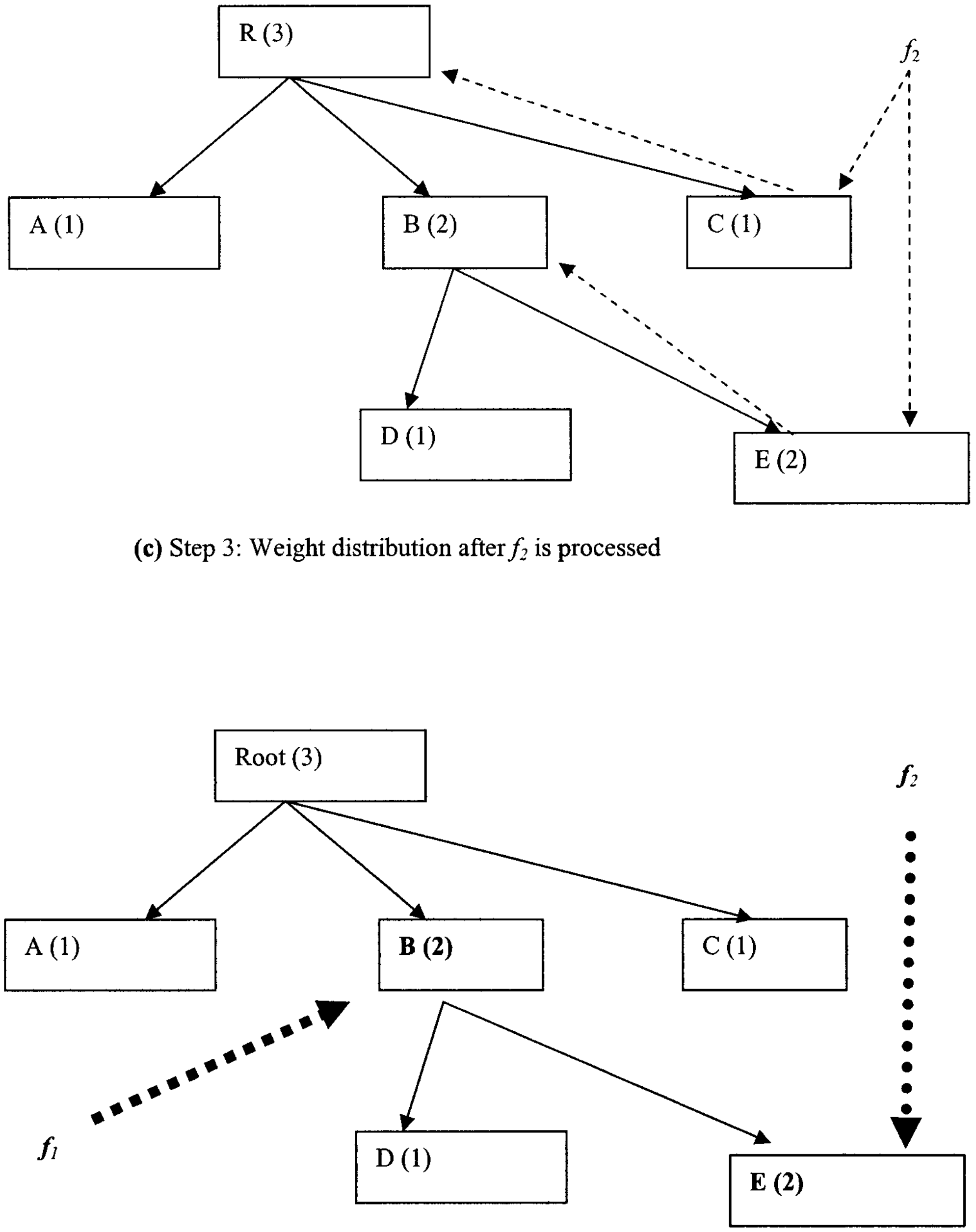
As a simple example (Figure 4.5), suppose two keywords f_1 and f_2 need to be disambiguated. For the first pass, suppose the keyword f_1 is mapped into synset *A* and *B*. As seen in Figure 4.5a, the weights in synsets *A* and *B* are increased by one, which is the number within the parentheses. In the second pass, because R is the hypernym of both A and B, the weight of R is increased by two. *D* and *E* are the hyponym synsets of B, so the weights of *D* and *B* are increased by one in Figure 4.5b. Then, f_2 is processed in a manner similar to that shown in Figure 4.5c. The last step (Figure 4.5d) is checking the most related synsets for f_1 and f_2. As a result, synset *B*, which has higher weight compared to *A*, is chosen for f_1 and synset *E* is chosen for f_2.



(a) Step 1: Word f_1 *is* mapped to the synsets in WordNet



(b) Step 2: Increase the weights of hypernyms and hyponyms



(c) Step 3: Weight distribution after f_2 is processed

(d) Step 4: Picking synset with maximum weight for each keyword

Figure 4.5 An Example of the Sense Disambiguation Method

After the sense disambiguation method mentioned above, for a document, each keyword is mapped into a single synset in the WordNet database, and the number of occurrences for each keyword is recorded. Figure 4.6 shows the simple example of a document containing only two keywords f_1 and f_2. The number of occurrences for each keyword is displayed in the brackets.

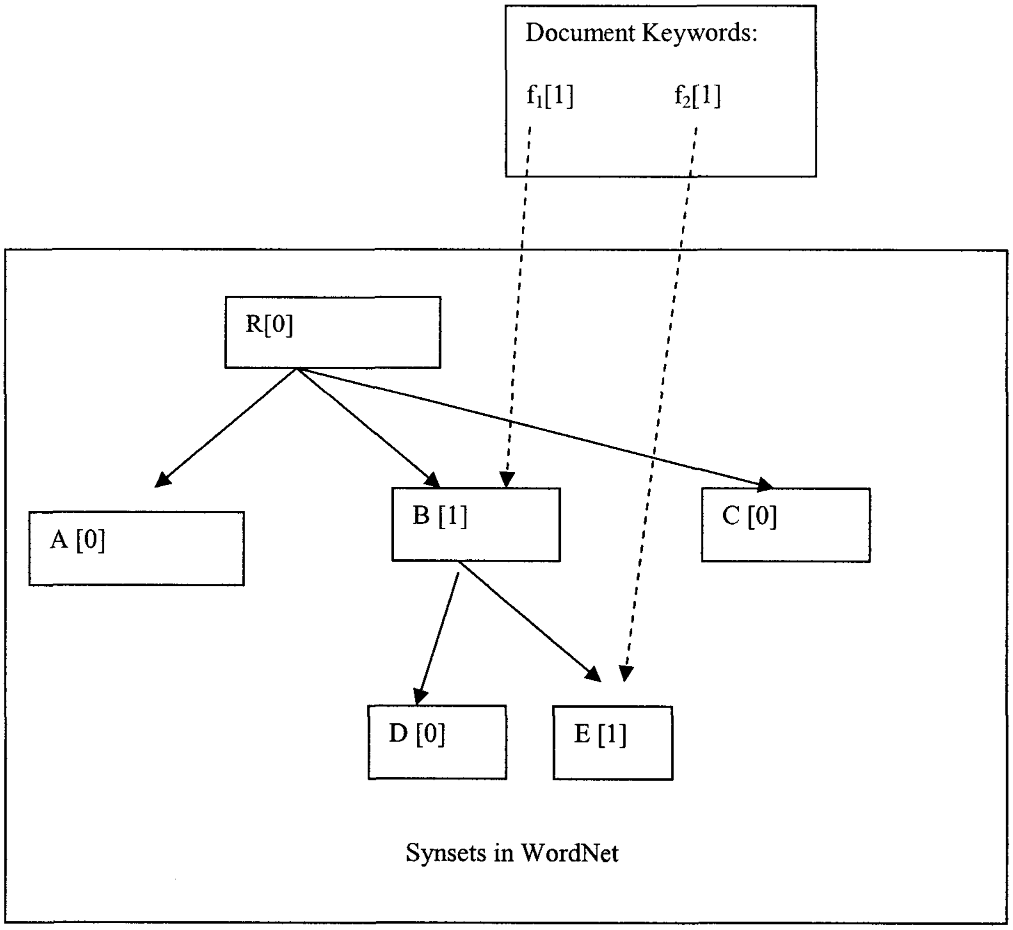


Figure 4.6 Mapping the Keywords to Synsets

As the last step, we calculate the probability of each synset by dividing the number of occurrences of this synset into the total number of occurrences of all the synsets. For example, the document shown in Figure 4.6 will be represented by two synsets, *E* and *B*, with a probability of 0.5 each.

4.4 Constructing the Semantic Hierarchy Representation

After the disambiguated sense mapping process, a document is represented by a list of synsets associated with a probability. These synset probabilities are calculated merely from the occurrences of the document keywords; therefore, these synset probabilities do not contain any information on the relationships between synsets. In Section 3.3, we describe a way of turning a group of synsets describing the category into the semantic hierarchy representation construction. Similar to the semantic hierarchy construction for category, we defined the original weight for each synset as the probability obtained in Section 4.3. Then, Formula 3.1 and Formula 3.2 are applied to calculate the propagated weights of each synset accordingly. After the propagation process, the document is represented in a distribution of propagated weights in the noun synset hierarchy of WordNet, which is the semantic hierarchy representation.

4.5 Classifying a Document

To classify a document, we calculate the similarity of the document to each category and select the category with maximum similarity for the document. Figure 4.7 lists the pseudo code for the classification algorithm.

```
Given a document in semantic hierarchy representation
For each category in the category hierarchy
    Calculate the similarity of the category and the document
Classify the document to the category with the maximum value
```

Figure 4.7 The Classification Algorithm in Pseudo Code

To evaluate the similarity *Sim(c_k,d)* between the *k*-th category c_k in a category hierarchy C and a document *d* having semantic hierarchy representation, we use the similarity measure listed in Formula 4.1.

$$\boxed{Sim(d,c_k)=\frac{\sum_{l=1}^{n}(d_l c_{k,l})}{\sqrt{\sum_{l=1}^{n}d_l^2}\sqrt{\sum_{l=0}^{n}c_{k,l}^2}}}$$

Formula 4.1 The Similarity Measure of a Document and a Category

In Formula 4.1, *n* is the number of synsets in the WordNet noun database (currently n= 79,689 [wnstat]). $c_{k,l}$ and d_l are defined as the propagated weights of the corresponding synset *l* in the semantic hierarchy representation of the document and category, respectively. Then, after checking all the categories in the category hierarchy *C*, the document *d* is classified to the category c_{max}, which has the maximum similarity value with the document, as in Formula 4.2.

$$\boxed{Sim(c_{\max},d)=\underset{c_k\subseteq C}{Max}(Sim(c_k,d))}$$

Formula 4.2 Choosing the Maximum Similarity

4.6 Algorithm Analysis

We used Figure 4.8 to illustrate the presented semantic hierarchy classification method, which includes three processes: sense disambiguation, semantic hierarchy construction, and document classification. We define the comparing operation on a keyword or a synset as the basic operation.

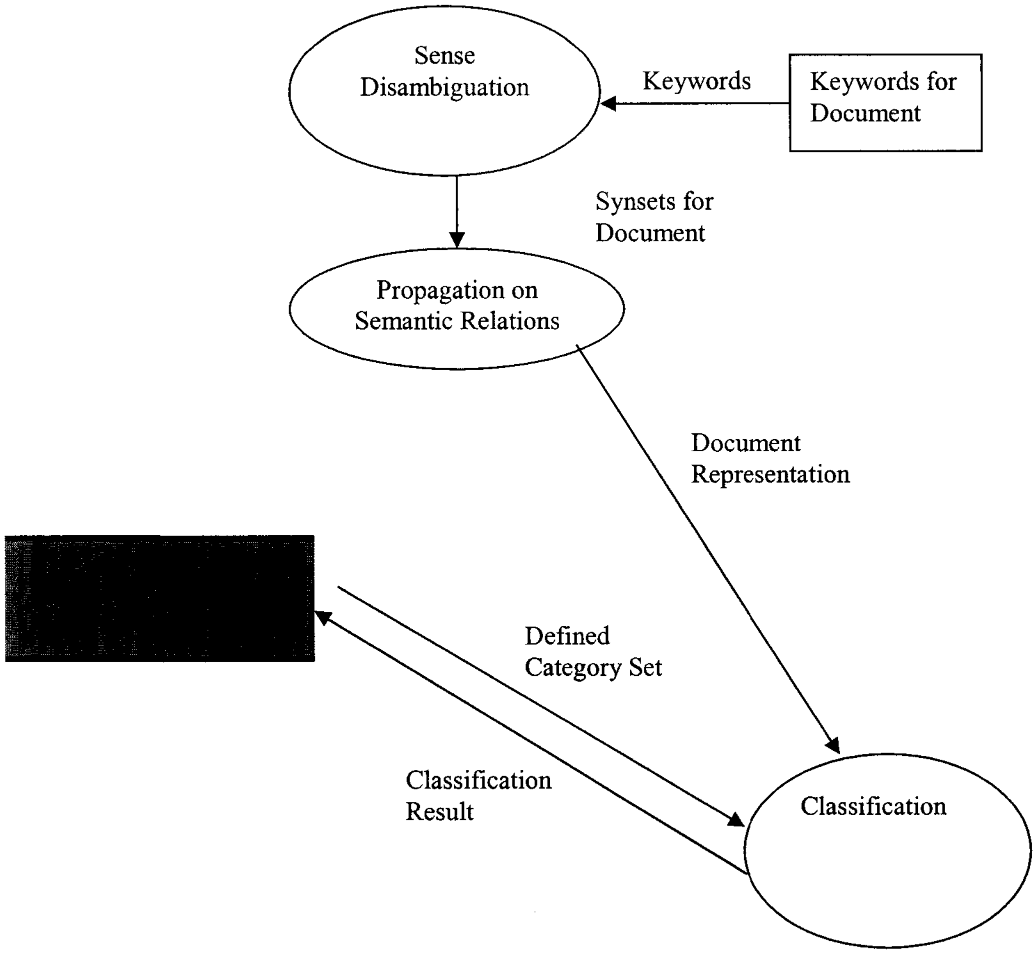


Figure 4.8 The Semantic Hierarchy Classification Algorithm

In the sense disambiguation process, let the number of keywords in a document be *k* and the maximum number of the hypernyms and hyponyms for a synset in the WordNet database be *h*. The worst case complexity for sense disambiguation is then *h* times *k*. As *h* is a constant number, so the complexity for the sense disambiguation is *O(k)*.

In the semantic hierarchy construction process, the worst case happens when these *k* keywords are mapped to the synsets that are in the deepest level in WordNet. Suppose the maximum number of levels of the synset hierarchy in WordNet database is *v*; then, each of them needs to be propagated by *v* times, and v is a constant, so the worst case complexity for this process will be O(k).

In the classification process, let the number of noun synsets in WordNet be *n*; then, the complexity of calculating the similarity of a category and a document will be O(*n)*. Suppose the predefined category hierarchy contains *u* categories; then, in order to get a classification result for the document, we have to compare the document to *u* categories in the worst case, which causes the worst case complexity for the classification process to be *O(un)*.

CHAPTER 5

CATEGORY EXPANSION BY CLUSTERING

5.1 Introduction

When the number of web pages that need to be classified grows, in order to achieve a better performance, the category hierarchy must necessarily develop more classes to accommodate all the pages. In order to achieve this goal, we use a new clustering approach that focuses on extracting similar web pages within a category to generate a new subcategory.

In Section 5.2, we present a category-based clustering algorithm to create additional categories when needed. Then, in Section 5.3, we provide a way to describe the new category. We generate a representation for the newly created category. This representation is based on the semantic hierarchy representation. We also provide a way to name the category in this section. The detailed process of each section is described in Figure 5.1

5.2 Category-based Clustering Method

In this section, we describe a category-based clustering method for creating a new category. The solution for creating a new category will be clustering a group of documents that are similar to each other while different from the category. For this purpose, we present the category-based clustering method. We also develop a new similarity measure, which is called the category-based clustering score (CBCS), by measuring the similarities between documents as well as the distinction from the category. Then, a maximum spanning tree method is applied to obtain a cluster based on these scores. The process is listed as pseudo code in Figure 5.2.

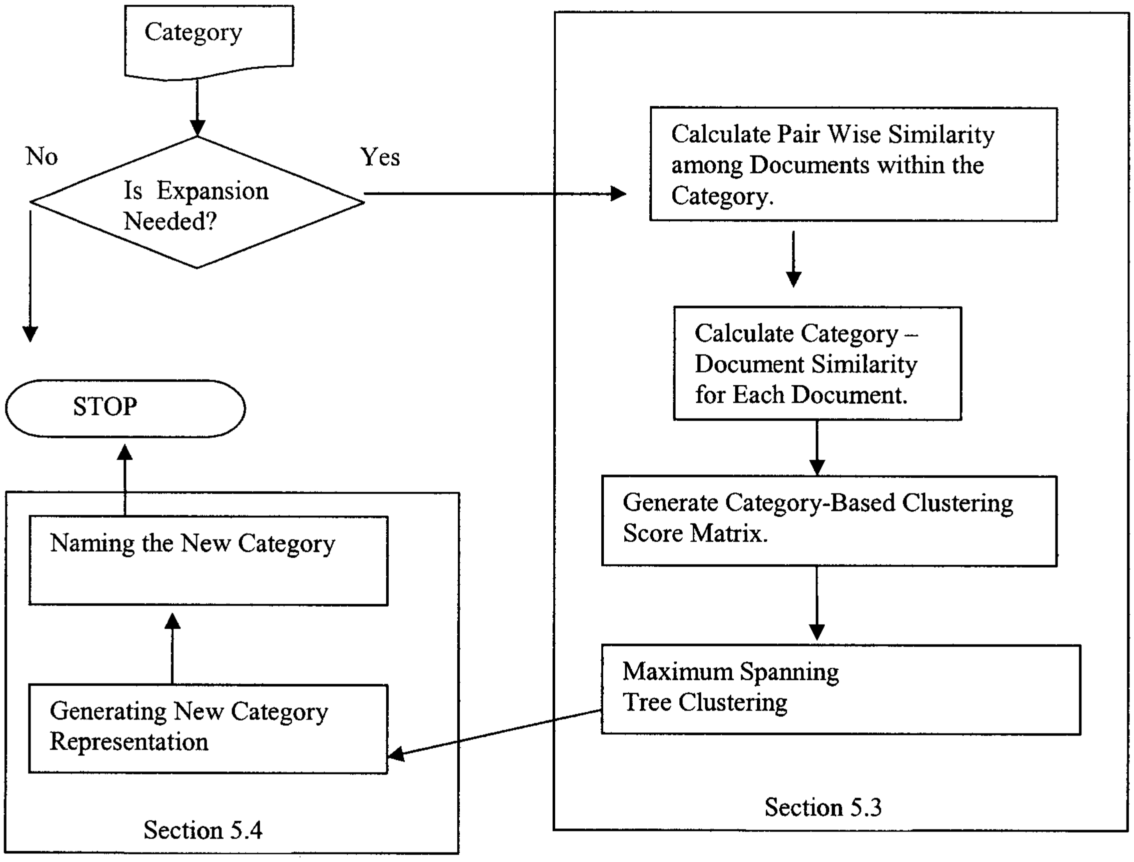

Category
No
Yes
Is Expansion Needed?
STOP
Naming the New Category
Generating New Category Representation
Section 5.4
Calculate Pair Wise Similarity among Documents within the Category.
Calculate Category – Document Similarity for Each Document.
Generate Category-Based Clustering Score Matrix.
Maximum Spanning Tree Clustering
Section 5.3

Figure 5.1 Category-Based Clustering Algorithm

In a category *c,* let D be a set of m documents, we construct an *m* by *m* matrix. The value in the i-th row and j-th column is the category-based clustering score of document d_i and d_j. The similarity $Sim(d_i,d_j)$ for two documents d_i and d_j is calculated by Formula 5.1. Let $Sim(d_i,c)$ be the similarity of a document d_i to the category *c*. The similarity is calculated following Formula 5.2. After all the similarities between the category and the documents are calculated, we extract the minimum category-document similarity $\underset{d_k \subset d}{Min}(Sim(d_k, c))$. Then, the category-based clustering score between the documents d_i and d_j with respect to category c is defined as *S(d* $S(d_i,d_j,c\ )$in Formula 5.3. In Formulas 5.1, 5.2, and 5.3, *n* is the total number of all the synsets in the semantic hierarchy representation. These synsets are labeled from 1 to *n*.

```
Retrieve category information
For each documents in the category that need to be expanded
    Calculate the similarity between the document and the category
For each pair of documents in the category that need to be expanded
    Calculate the similarity between two documents by Formula 5.2
    Calculate the category-based clustering score for the pair of documents by Formula 5.3
Use  maximum spanning tree for clustering
Name the new cluster
```

Figure 5.2 The Category-Based Clustering Algorithm

$$Sim(d_i,d_j) = \frac{\sum_{l=1}^{n}(d_{il}d_{jl})}{\sqrt{\sum_{l=1}^{n} d_{il}^2}\sqrt{\sum_{l=0}^{n} d_{jl}^2}}$$

Formula 5.1 Similarity of Two Documents

$$Sim(d_i,c) = \frac{\sum_{l=1}^{n}(d_{il}c_l)}{\sqrt{\sum_{l=1}^{n} d_{il}^2}\sqrt{\sum_{l=0}^{n} c_l^2}}$$

Formula 5.2 Similarity of a Document with a Category

$$\boxed{S(d_i, d_j, c) = \frac{Sim\ (d_i, d_j)}{\log(\frac{(Sim\ (d_i, c) + Sim\ (d_j, c))}{\underset{d_k \subset D}{Min}\ (Sim\ (d_k, c))})}}$$

Formula 5.3 CBC Score of Document *di dj* in Category *c*

To avoid the division by zero error in Formula 5.3, our system ensures that a document d_j will not be put into a category c if the similarity is zero. Because of this, the minimum similarity value $\underset{d_k \subset D}{Min}(Sim(d_k, c)$ is larger than zero. To another extreme, if the case that $Sim(d_i,c)$ =1 happens, which means that document d_i is equal to the category description, then document d_i should remain in the original category and, therefore, excluded for the clustering process.

We modify a maximum spanning tree clustering algorithm, as found in Asano et al. [1988], and apply it in our clustering process. In order to retrieve a cluster of documents that meets the requirement for category expansion, we treat each document as a vertex in a graph. The weight for each edge of the graph connecting two vertexes is defined as the category-based clustering score between the two documents represented by the vertex.

The clustering algorithm starts by picking the edge with a maximum weight. The two vertices connected by this edge form the initial cluster. Then, the growth of the cluster follows the rules listed below:

(1) Select the edge that has the maximum weight between a clustered and non-clustered vertex.

(2) Add the selected edge and non-clustered vertex to the cluster.

(3) If the stopping criteria are met, stop clustering; else go to step (1).

Generally, at least two ways to stop the clustering process are possible. The first one is setting a threshold for the similarity. This approach usually calculates the average similarity of all the documents within the newly created cluster and chooses a threshold to stop clustering when the similarity is below the threshold. The other way is by limiting the size of the cluster. When the number of the documents in the cluster reaches the threshold, we stop the clustering algorithm. We choose to use the size threshold. For experimental purposes, we have chosen several different cluster sizes to test the performance of this category-based clustering method, and the results of these experiments are presented in Chapter 6.

5.3 Generating a Representation for New Category

The way to generate the representation for the new category is by incorporating the information from all the clustered documents.

Let q be the number of documents in the new category. Each of the documents is in a semantic hierarchy representation, which has n synsets labeled from 1 to n. The semantic hierarchy representation for a new category is generated with two steps. In the first step, the weight w_i' for each synset i in the semantic hierarchy for the new category is created by summing the propagated weights $w_i^{(j)}$ of corresponding synsets in each of the clustered documents according to Formula 5.4.

$$w_i' = \sum_{j=1}^{q} w_i^{(j)}$$

Formula 5.4 The Sum of m Propagated Weights

The second step is normalizing the weight of each synset in the new semantic hierarchy representation according to the formula $w_i'' = \frac{w_i'}{\sum_l w_l'}$.

After the new category is created, the name is necessary for referencing the new category. As the final step, we select the first word in the synset with the highest weight of the new category representation as the category name.

5.4 Algorithm Complexity

In order to analyze the computational complexity of the category-based clustering algorithm, we define a basic operation as the comparing operation on documents. Let *m* be the number of documents in the category that need to be expanded. To calculate the pairwise document similarities, we need to compare m(m-1)/2 times. The complexity for this step is $O(m^2)$. Computing the similarity between a category and each document requires *m* comparison; therefore, the complexity will be $O(m)$. In order to calculate the category-based clustering score for each pair of documents, we use the similarity of this pair of documents as well as two similarities of these two documents to the category, respectively. Because the category-based clustering score matrix is an *m* by *m* matrix, we need to have m^2 category-based clustering scores. The complexity for constructing this matrix is then $O(m^2)$. The last step is clustering documents using a maximum spanning tree algorithm. Considering each time one document is included into the cluster, the complexity for the maximum spanning tree is $O(m)$. So, the overall complexity of the category-based clustering method is $O(m^2)$.

CHAPTER 6

EXPERIMENTS AND RESULTS

In this chapter, we design experiments to test the methods related to the semantic hierarchy classification system. The experiments in Section 6.1 are designed for selecting sources to represent the category. In Section 6.2, experiments are designed to evaluate the performance of the classification system in a category system. In Section 6.3, we compare our classification method with other related classification methods by using the information in two categories of Usenet. Finally, in Section 6.4, two experiments are designed to test the category-based clustering algorithm for category expansion.

6.1 Experiment on Sources for Category

We use a section of Yahoo's [Yahoo] category system as the test base to conduct the first experiment. The structure of the categories is shown in Figure 6.1. In this experiment, the impact of using two different sources to describe the category description is tested: the first source, as found in Labrou and Finin [1999], is using the keywords of the summaries and titles (ST) for the web pages provided in Yahoo categories. The other source to describe the category is using category names and meronyms (NM) as described in Chapter 3.

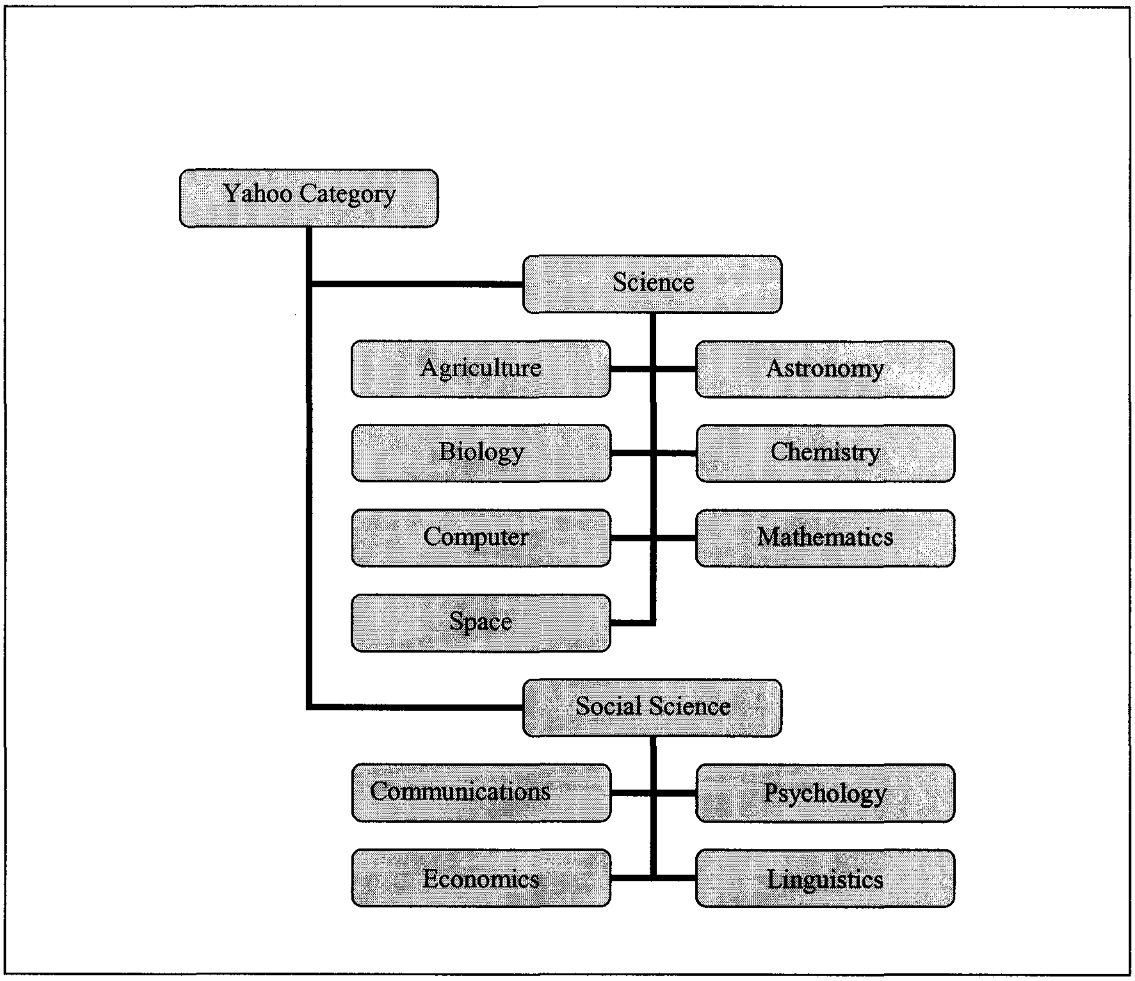


Figure 6.1 Yahoo Category Structure Used for the Experiment

To compare the two different sources describing the category, we use 200 pre-classified web pages taken from the two-level category hierarchy in Yahoo's structure. These web pages are then sent to the semantic hierarchy classification with two difference sources. The results of the experiments are listed in Table 6.1, Figure 6.2, and Figure 6.3. Labels in Table 6.1 are defined as follow: *Correct* stands for the result that a web page is classified to the category where it is taken. *Not Deep Enough (NDE)* means a web page is classified to the parent category of its original category. On the other hand, if the classification result for a web page is the child category of the original category, it is called *Expanded.* The category hierarchy is a two-level hierarchy. In the levels where the misclassified cases occur, we label them with *Error* in different levels. For example, if a web page, originally taken from the Biology under Science category, is classified to any subcategory within Social Science, this error is called Error in 1st Level. If a web page is classified to any other subcategory within the Science category this error is called Error in 2nd Level.

Table 6.1 Classification Results on Hierarchy Category Structure in Number

	ST	NM
Error in 1st Level	29	20
Error in 2nd level	109	42
Expanded	2	0
Not Deep Enough (NDE)	26	76
Correct	34	62

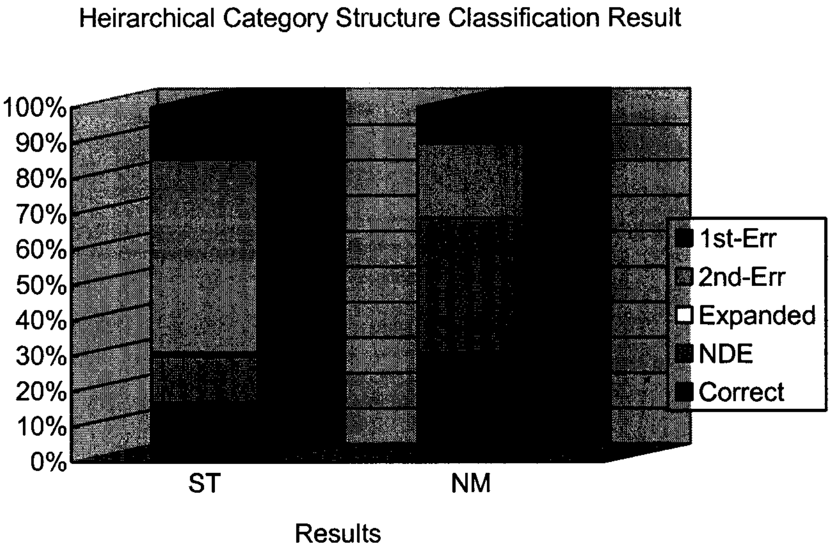


Figure 6.2 Classification Results on Hierarchical Category Structure in Percentage

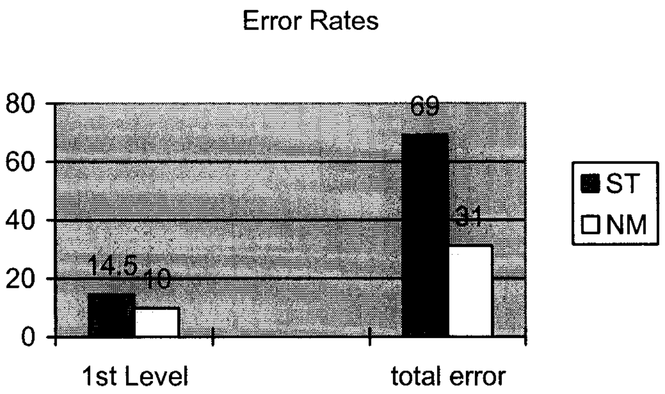


Figure 6.3 Error Rates on Hierarchical Category Structure in Percentage

In this experiment, we are most concerned about the result that is indicated as "Correct." The experiment shows that our method of using the name of the category and meronyms (NM) performs better than the existing method (ST).

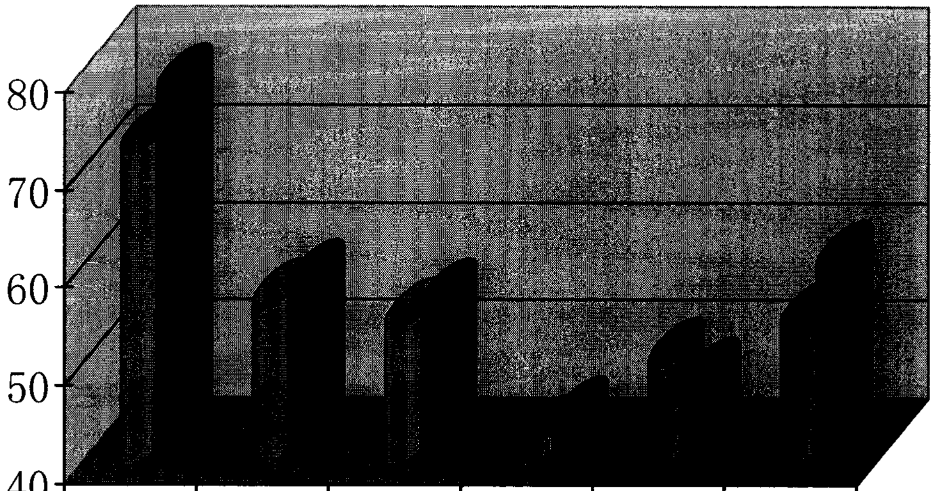
6.2 Comparison on Keyword-Based Classification

To evaluate our semantic hierarchy classification system, we design an experiment in this section to compare it with an existing keyword-based classification system presented in our earlier paper [Peng and Choi 2002]. Ten categories from the Yahoo Category Structure [Yahoo] are chosen for testing purposes as listed in Table 6.2.

Table 6.2 Category Setting of One Level Category Experiment

Categories	**URLs**
Health	http://dir.Yahoo.com/Health/
Science	http://dir.Yahoo.com/Science/
Government	http://dir.Yahoo.com/Government/
Business	http://dir.Yahoo.com/Business_and_Economy/
Education	http://dir.Yahoo.com/Education/
Movies	http://dir.Yahoo.com/Entertainment/Movies_and_Film/
Art	http://dir.Yahoo.com/Arts/
Religion	http://dir.Yahoo.com/Society_and_Culture/Religion_and_Spirituality/
Sports	http://dir.Yahoo.com/Recreation/Sports/
Social Science	http://dir.Yahoo.com/Social_Science/

Although the system has ten categories, the testing examples are web pages taken from six of the Yahoo categories mentioned on Table 6.2: Science, Business, Government, Religion, Sports, and Social Science. The remaining four categories, Art, Health, Movie, and Education, are used only to increase the difficulty of the classification task. We randomly selected fifty web pages from each of the six categories, and Figure 6.4 records the experiment results from two classification systems.



	Science	Business	Government	Religion	Sports	Social Science
▨ Keyword	72	56	54	42	50	54
■ Sense	78	58	56	44	48	60

Figure 6.4 Results for Accuracy of One Level Category Hierarchy Experiment

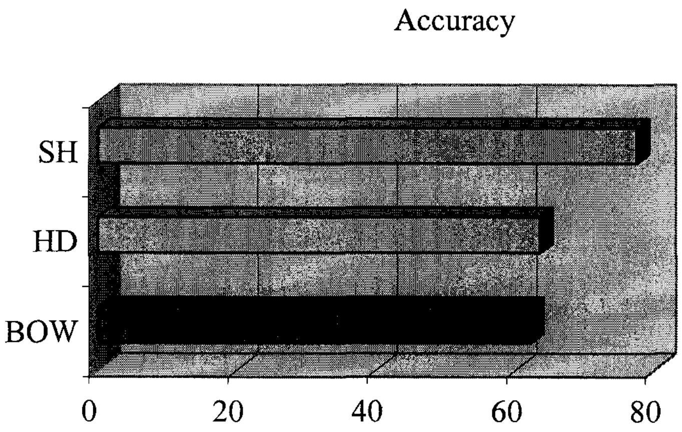
The classification results are provided in Figure 6.4. Based on the results, the sense-based classification shows a 6% improvement in both the Science category and the Social Science category. In Business, Government, and Religion categories, the classification results show an improvement of only 2%. Especially in the Religion category, using keyword-based classification produces a better performance.

The reason for these results might be that some of the keywords for describing the categories are proper names, such as web site names or brands. However, those names are not in the WordNet database and are omitted by the sense-based classification system. The absence of these keywords reduces the accuracy of the sense-based classification.

6.3 Comparison of Other Sense-Based Classification Systems

In this section, we design another experiment to compare our semantic hierarchy classification system with other sense-based classification systems. Usenet is a discussion system, which consists of a set of "newsgroups" with names that are classified hierarchically by subject. Different terminology, varied in topic and special writing style, makes the classification task on Usenet very difficult. Scott and Matwin [1998] performed an experiment on two news groups in Usenet: bionet.microbiology and bionet.neuroscience. They compared the sense-based classification system using “hypernym density” with the existing keyword-based classification using “Bag of Words” and shows little improvement. For comparison, we choose the same newsgroups to test the semantic hierarchy classification system.

We obtain our testing examples from Usenet by using the Google Groups website [Google Groups]. At the date of our experiment, approximately 21,000 postings are in the bionet.microbiology newsgroup and 34,700 postings in the bionet.neuroscience news group. We randomly select 217 postings (98 in microbiology and 119 in neuroscience) from two newsgroups as the testing example. The testing results of our “Semantic Hierarchy” (SH) classification system are summarized in Figure 6.5. The experimental results of using “Bag-of-Words” (BOW) and “Hypernym Density” (HD) classification systems are taken from Scott and Matwin [1998].



	BOW	HD	SH
■ Accuracy	62.14	63.57	77.46

Figure 6.5 Comparisons on Accuracy Rates in Usenet Newsgroups

From Figure 6.5, we can see that our semantic hierarchy classification system achieves a highest accuracy rate of 77.46% while the other two systems only achieve accuracy rates at around 63%. The improvement of the semantic hierarchy system is more than 13%.

6.4 Experiment on Category-Based Clustering

In this section, the performance of the category-based clustering is tested. For testing purposes, we use the Science category and six subcategories (Agriculture, Astronomy, Biology, Chemistry, Computer, Mathematics, and Space) in Yahoo as the predefined category hierarchy. The testing examples are web pages taken from another category, Physics, which are not in the original category hierarchy. The category hierarchy as well as the additional testing category are listed on Figure 6.6. We design a set of experiment to test the performance of the category-based clustering algorithm in section 6.4.1. In section 6.4.2, another set of experiments is designed to test the effectiveness of the representation of a newly created category.

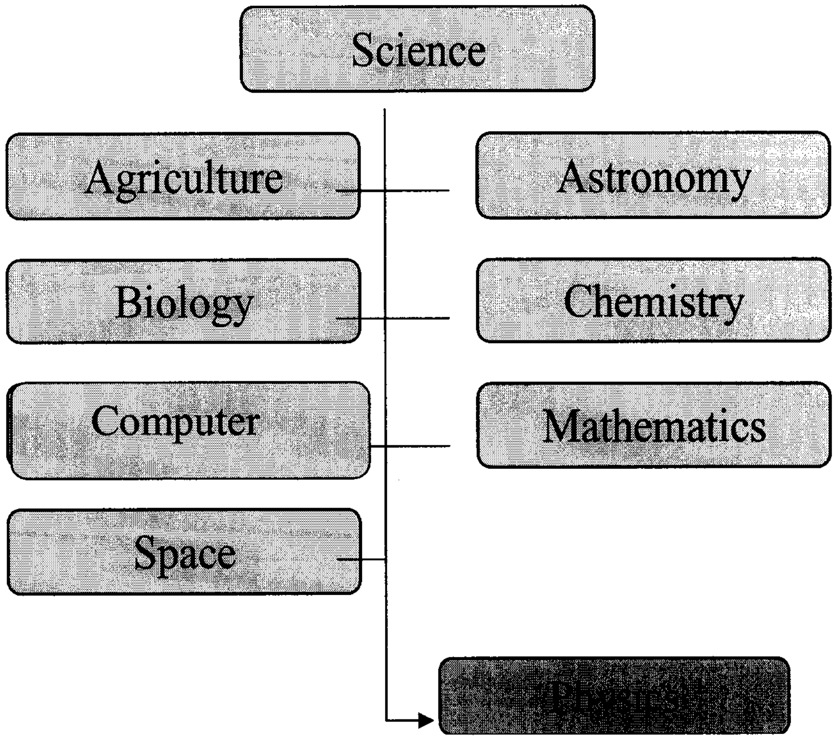
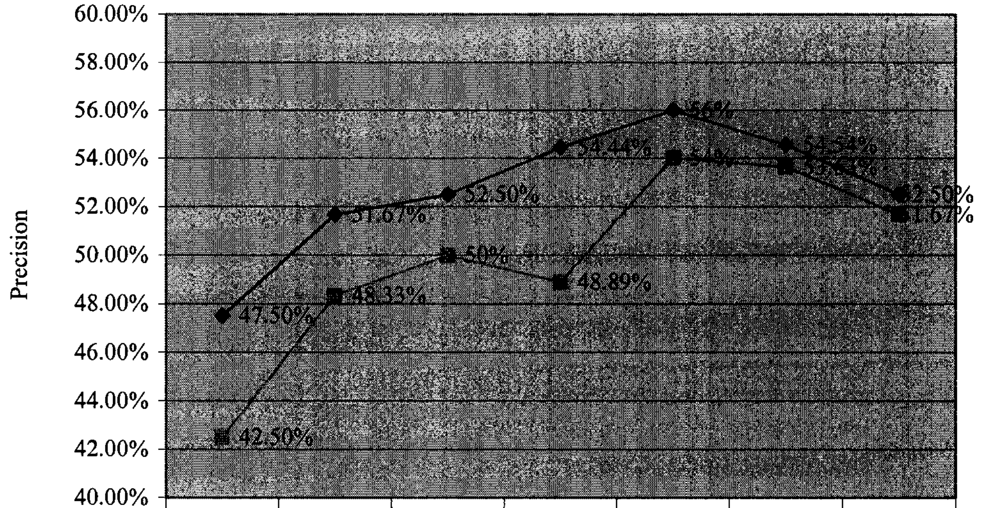


Figure 6.6 Category Setting for Category-Based Clustering

6.4.1 Clustering Performance

To evaluate the clustering performance, we compare the category-based clustering score (Formula 5.3) with the similarity measure (Formula 5.1) that does not take advantage of the category information. We randomly select 100 web pages taken from the category "Physics," as the testing examples, because the "Physics" category is not contained in the original predefined category hierarchy. The best place to host all of these testing examples is in the "Science" category. We use the semantic hierarchy classification system to classify these web pages, and 70 web pages are found in the expected "Science" category. The rest of the testing examples are misclassified to other categories originally in the system. Then, these 70 web pages, combined with the 76 web pages predicted “not deep enough” in the experiment of Section 6.1, are used to test the category-based clustering method.

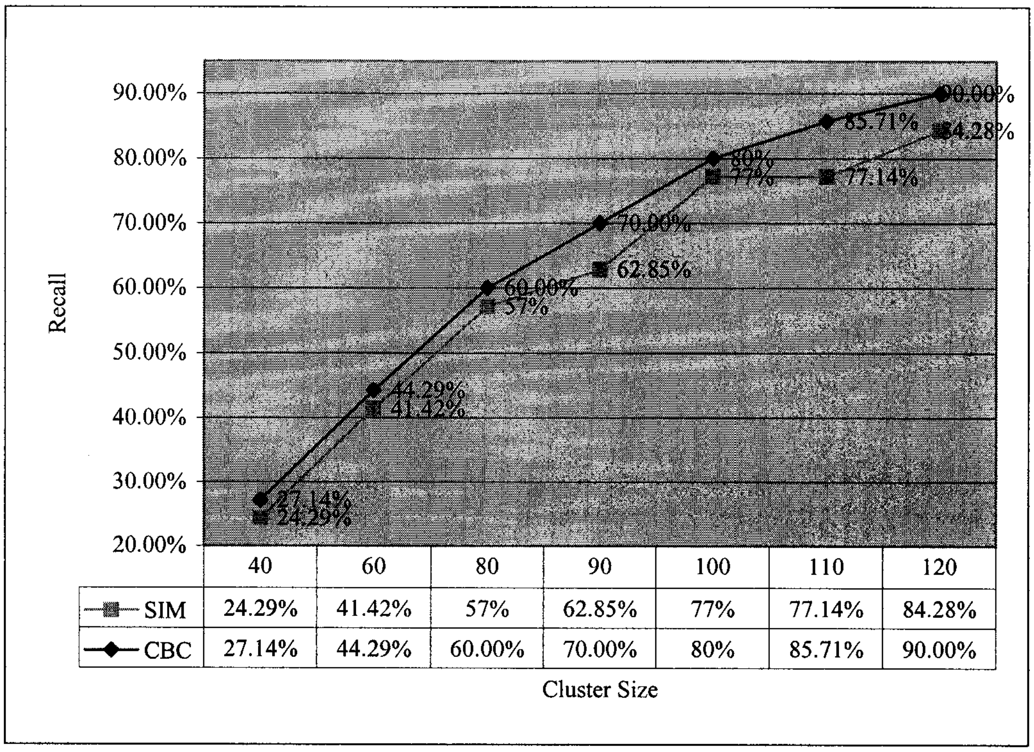
The results are shown in Figures 6.7, 6.8, and 6.9 concerning three different clustering performance measures: precision, recall and the F1-value. We compare the category-based clustering (CBC) method with the method that uses only similarities between web pages (SIM) and set the thresholds according to the cluster size. The seven results are taken at points with the cluster sizes of 40, 60, 80, 90, 100, 110, and 120.



	40	60	80	90	100	110	120
—■— SIM	42.50%	48.33%	50%	48.89%	54%	53.63%	51.67%
—◆— CBC	47.50%	51.67%	52.50%	54.44%	56%	54.54%	52.50%

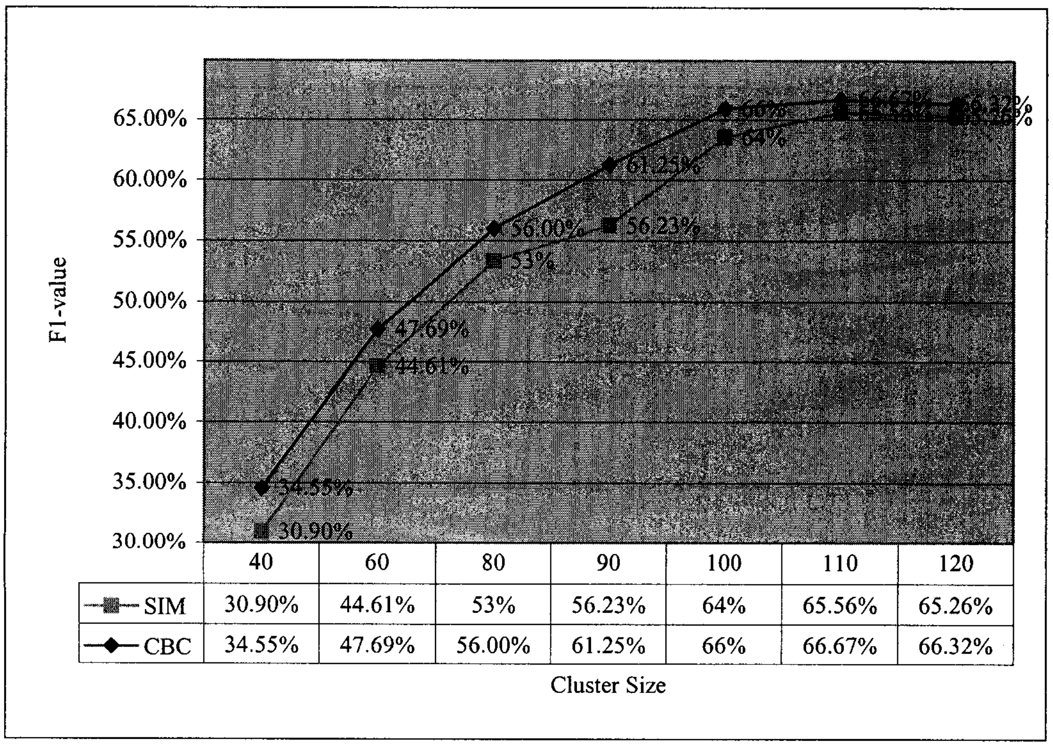
Cluster Size

Figure 6.7 Comparisons of Precision Results of Two Cluster Measures



	40	60	80	90	100	110	120
SIM	24.29%	41.42%	57%	62.85%	77%	77.14%	84.28%
CBC	27.14%	44.29%	60.00%	70.00%	80%	85.71%	90.00%

Figure 6.8 Comparisons of Recall Results of Two Cluster Measures



	40	60	80	90	100	110	120
SIM	30.90%	44.61%	53%	56.23%	64%	65.56%	65.26%
CBC	34.55%	47.69%	56.00%	61.25%	66%	66.67%	66.32%

Figure 6.9 Comparisons of F1 Value Results of Two Cluster Measures

From the testing results in Figure 6.7, Figure 6.8, and Figure 6.9, we can see that the category-based clustering measures have a better performance with an average improvement of around 3% in these three clustering measures.

To evaluate the performance of the category-based clustering method further, we conduct another experiment using the “Geography” category in Yahoo. Similar to the experiment on Physics, we test 100 web pages randomly selected from the “Geography” category and discover that there are 62 web pages classified to “Science.” Then, we apply the category-based clustering method and the method that uses only similarities between web pages on a total number of 138 web pages (62 web pages that are classified to Science and 76 that predicted “not deep enough” in the experiment of Section 6.1). The thresholds are set according to the cluster size. Table 6.3 records the precision values concerning the difference of the two measures. The precision values are recorded at points with the cluster sizes of 40, 50, 60, 70, 80, 90, and 100.

Table 6.3 Results of Precision of Two Methods in Geography Category

Cluster size	Precision SIM	Precision CBC
40	70%	82.5%
50	74%	84%
60	75%	78.33%
70	67.14%	70%
80	58.75%	61.25%
90	57.78%	60%
100	53%	54%

The results in Table 6.3 show that the CBC method out-performs the SIM method in all testing points. The average improvement of using CBC method in this experiment is about 5%.

6.4.2 Tests on Effectiveness for New Category Representation

We have designed the following experiment to test the effectiveness of the presented method of generating representation for the new category. After the clustering, the new category "Physics" is added to the classification system. We use 60 testing web pages in this experiment. In the testing web pages, 30 web pages are the misclassified web pages (M30) in the Physics category before the category is added. The other 30 web pages are randomly selected from the existing categories in the original category hierarchy (R30). Results of this experiment are listed in Table 6.4.

Table 6.4 Testing of Effectiveness of the Newly Generated Category Information

	Correct Classification Result	Wrong Classification Result	Accuracy
M30	29	1	96.67%
R30	23	7	76.67%

From the results in Table 6.4, the presented method of generating the representation shows high effectiveness with an average accuracy of over 85%.

CHAPTER 7

SUMMARY AND PROSPECTS

7.1 Summary

This dissertation presents a system that automatically classifies documents based on the meanings of words and the semantic relationships between these meanings. To classify a document, the system extracts keywords occurring in the document and maps them to synsets defined in the WordNet database after sense disambiguation. The original weight of each synset is calculated and then propagated to its related synsets according to the sematic hierarchy in WordNet. After propagation, the semantic hierarchy information provided by WordNet is captured by a distribution of propagated weights on the synsets, which is a new semantic hierachy document representation. The classification algorithm is based on the similarity of a document and a category in the same document representation. A document needing to be classified is then compared to all the categories. The category with the most similarity to the document is chosen as the host for the document. Comparing to previous experiments on the Usenet data, the semantic hierarchy classification approach increases the classification accuracy by 13%.

The experimental results of selecting different sources for a category in Yahoo indicate that using difference sources for a category has a significant effect on the overall accuracy of the semantic hierarchy classification system. In particular, the method of using the name of the category and its meronyms achieves a significant improvement compared to the related method of using the title and the description of the category. However, we should point out that this result reflects the specific need for the semantic hierarchy classification and may not be applicable for other systems that do not take advantage of word senses.

The system also addresses the problem of having a fixed number of categories, which is ignored by most classification systems, and provides a solution by using a category-based clustering method. When a category expansion is needed, pair wise similarities of the documents as well as the similarities of each document to the category are calculated. Then, these two kinds of similarities are used to calculate category-based clustering scores. Based on the scores, a maximum spanning tree algorithm is applied to capture the document cluster that is far from the category. Comparing with the method using normal similarity measure, the category-based clustering method performs better in three key factors of clustering: precision, recall, and F1 measure. The highest improvement, which is more than 7%, appears in recall in our experiment.

7.2 Future Study and Prospects

This dissertation provides a way for the future of applying semantics for classifications. It also shows that relationships between groups of meanings or concepts are promising sources for mining semantic information from documents. Much future work can be done in this direction on the move to the future of a semantic information age. For semantic classification, we are expecting the following future advances:

- An extension to use verbs, adverbs, and adjectives provided by the WordNet database, in addition to using nouns.

- An improvement in hierarchical classification using semantics.
- Classification on file formats other than text, for example jpeg files, swf files, and gif files by obtaining the content by machine in text form.
- An extension to multilingual systems by using different language versions of WordNet.

This dissertation also provides a new way of combining text classification and clustering. It challenges the idea of using clustering for classification and shows that classification can also help clustering for some special purposes. The future expectation would be the full discovery of the tight relationship between these two areas. The fact that classification will reduce the data size might be the key to applying a better clustering algorithm in the future. In addition, this clustering result will, in turn, benefit the area of classification by generating a more precise category description.

Finally, because the information online is growing rapidly and most of the information is not organized into meaningful categories, information retrieval is difficult. Classification on Internet resources is a good way to make users retrieve useful information easily. This dissertation provides a starting point for the coming classification standard. Applying the hierarchical structure as well as the dynamic growing mechanism in classification will help the development of the Internet in the future.

REFERENCES

Aas, K. and Eikvil, L. (1999) “Text Categorization: A Survey.” Report NR 941, Norwegian Computing Center.

Agirre, E. and Rigau, G. (1996) “Word Sense Disambiguation Using Conceptual Density.” In Proceedings of COLING'96.

AltaVista, http://altavista.digital.com

Ananiadou S. and Tsujii, J. (1997) “Term Disambiguation by Adding Structural Constraints to Lexically Based Context Matching Techniques.” In Proc. of NLPRS'97, Phuket, Thailand.

Ananiadou, S. (1996) “Towards a Linguistic Treatment of Compounds in a Machine Translation Environment,” in Journal of Natural Language Processing,Vol.3, No.1, pp.45-66.

Apte, C., Damerau, F. and Weiss, S. (1994) “Towards Language Independent Automated Learning of Text Categorization Models.” In Proceedings of the 17th Annual ACM/SIGIR conference.

Asano,T., Bhattacharya, B., Keil, M., and Yao, F. (1988) “Clustering Algorithms Based on Minimum and Maximum Spanning Trees.” In Proceedings of the 4th Annual Symposium on Computational Geometry, pages 252--257, June.

Attardi, G. M. Simi, F. Tanganelli, A. Tommasi. (1999) “Learning Conceptual Descriptions of Categories.” Rapporto Tecnico, Dipartimento di Informatica, TR-99-21.

Baez, John C., Dolan, James. (1995) “Higher-Dimensional Algebra and Topological Quantum Field Theory.” Jour. Math. Phys. 36, 6073-6105.

Bar-Hillel, Yehoshua (1960) “Automatic Translation of Languages.” In Alt, Franz; Booth, A. Donald and Meagher, R. E. (Eds), Advances in Computers, Academic Press, New York. 247-261.

Barzilay, R. and M. Elhadad (1997) “Using Lexical Chains for Text Summarization.” In ACL/EACL Workshop on Intelligent Scalable Text Summarization, pages 10-17.

Beckwith R. and Miller G. A. (1990). "Implementing a Lexical Network." In International Journal of Lexicography 3 (4), 1990, pp. 302 – 312

Berkhin, P. (2002) "Survey of Clustering Data Mining Techniques." Technical Report, Ac- crue Software, 2002. 3.

Brunk, C., and Pazzani, M. (1995). "A Linguistically-Based Semantic Bias for Theory Revision." In Proceedings of the 12th International Conference of Machine Learning.

Chan, Philip K. (1999) "A Non-Invasive Learning Approach to Building Web User Profiles." KDD-99 Workshop on Web Usage Analysis and User Profiling.

Chekuri, C., Goldwasser, M., Raghavan, P., and Upfal, E. (1996) "WebSearch Using Automatic Classification." In Proceedings of the Sixth International World Wide Web Conference.

Choi, Ben, and Peng, Xiaogang (2004) "Dynamic and Hierarchical Classification of Web Pages." Online Information Review, Vol. 28, No. 2, pp. 139-147.

Choi, Ben and Yao, Zhongmei (2004) Book Chapter "Web Page Classification." On Recent Advances in Data Mining and Granular Computing (mathematical aspects of knowledge discovery), Springer-Velag.

Cohen, W.W. (1995) "Text Categorization and Relational Learning." In Proceedings of the 12th International Conference in Machine Learning, pp.124-132.

Cohen,W.W. and Hirsh, H. (1998) "Joins that generalize: Text classification using WHIRL," in Proc. Of the Fourth Int. Conference on Knowledge Discovery and Data Mining.

Copestake, Ann and Lascarides, Alex (1997) "Integrating Symbolic and Statistical Representations: The Lexicon Pragmatics Interface." Proceedings of 35th Annual Meeting of the Association for Computational Linguistics. ACL Press, New Jersey.

Cortes, C. and Vapnik, V. (1995). "Support Vector Networks." Machine Learning, 20, pp. 273-297.

Cowie, Jim, Guthrie, Joe A., and Guthrie, Louise (1992) "Lexical disambiguation using Density." In Proceedings of COLING'96.

Dewdney, N., VanEss-Dykema, C., and MacMillan, R. (2001) "The Form is the Substance: Classification of Genres in Text." ACL Workshop on Human Language Technology and Knowledge Management.

Dhillon, Inderjit S. and Modha, Dharmendra S. (2001) "Concept Decompositions for Large Sparse Text Data Using Clustering." Machine Learning, 42(1):143-175, January.

Dhillon, Inderjit S., Fan, J., and Guan, Y. (2001) "Efficient Clustering of Very Large Document Collections." Data Mining for Scientific and Engineering Applications, Kluwer Academic Publisher.

Dominggos, P. and Pazzani, M. (1997) "On the Optimality of the Simple Baysian Classifier Under Zero-One Loss." Machine Learning 29, pp. 103-130.

Duda,R. O, Hart, P. E., and Stork, D. G. (2001) "Pattern Classification (2th Ed.)." Wiley, New York.

Dumais, S. T., Furnas, G. W., Landauer, T. K., and Deerwester, S. (1988) "Using Latent Semantic Analysis to Improve Information Retrieval." In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.

Dumais, S., Platt, J., Hecherman, D., and Sahami, M. (1998). "Inductive Learning Algorithm and Representations for Text Categorization." In Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management, pp. 148-155.

Everitt, B. S., Landua, S., and Leese, M. (2001) "Cluster Analysis." Arnold, London Great Britain,

Federici S., Montemagni S., and Pirrelli V. (1997). "Inferring Semantic Similarity from Distributional Evidence: an Analogy-Based Approach to Word Sense Disambiguation." In Proceedings of the ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications.

Fellbaum, C. (1990) "English Verbs as a Semantic Net." In International Journal of Lexicography, Vol 3, No.4 (Winter 1990), pp. 278-301.

Fellbaum, C. (1997) "A Semantic Network of English Verbs." in C. Fellbaum (ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Fellbaum, C. (1995) "Co-Occurrence and Antonymy." Journal of Lexicography, vol. 8-2, Oxford University Press.

Fellbaum, C. (1993) "English Verbs as Semantic Net," Journal of Lexicography, vol. 6, Oxford University Press.

Fellbaum, Christian, (1998) "The WordNet: An Electronic Lexical Database." MIT Press, "Map for Text Classification." In IJCAI-99 Workshop on Machine Learning for Information Filtering.

Frantzi, K.T., Ananiadou, S., and Tsujii, J. (1997) "Term Identification Using Contextual Cues." In Proc. of the 2nd Workshop on Multilinguality in Software Industry: the AI

Contribution (MULSAIC'97), at the International Joint Conference on Artificial Intelligence (IJCAI-97), August 23-29, Nagoya, Japan.

Galina Kan, (2000) “The Internet in Russia.” Working Paper eLab, Owen Graduate School of Management, Vanderbilt University.

Ganesh, Ramakrishnan, Prithviraj, B. P., Deepa, A., Pushpak, Bhattacharyya, and Soumen, Chakrabarti. (2004) “Soft Word Sense Disambiguation.” International Conference on Global Wordnet (GWC 04), Brno, Czeck Republic, January, 2004.

Gilarranz, J., Gonzalo, J., and Verdejo, M. (1997) “An Approach to Cross-Language Text Retrieval with the Eurowordnet Semantic Database.” In AAAI Spring Symposium on Cross-Language Text and Speech Retrieval. AAAI Spring Symposium Series.

Gonzalo, J., Verdejo, F., Peters, C., and Calzolari, N. (1998) “Applying EuroWordNet to Multilingual Text Retrieval.” in Computer and the Humanities, Special Edition on EuroWordNet, fc.

Goldstein, Jade, Kantrowitz, Mark, Mittal, Vibhu, and Carbonell, Jaime (1999) “Summarizing Text Documents: Sentence Selection and Evaluation Metrics.” In Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval (SIGIR-99), Berkeley, CA, pp. 121-128.

Google: http://www.google.com

Google Groups: http://www.google.com/grphp

Gravano, L., Garcia-Molina, H., and Tomasic, A. (1999) “Text-Source Discovery Over the Internet.” ACM Transactions on Database Systems, 24(2):229-264, June

Grefenstette Gregory, Schulze Maximilian, Heid Uli, Fontenelle Thierry and Gérardy Claire, (1996) “The DECIDE Project: Multilingual Collocation Extraction.” Proceedings of the European Association for Lexicography (EURALEX), Gothenburg.

Grossman, D., Frieder, Holmes, O. D., and Roberts, D. (1997) “Integrating Structured Data and Text: A Relational Approach.” Journal of the American Society for Information Science, 48(2).

Guthrie, J., Guthrie, L., Wilks, Y., and Aidinejad H., (1991) “Subject-Dependent Co-Occurrence and Word Sense Disambiguation.” ACL-91, pp. 146-152.

Hirst, G. and St-Onge, D. (1997) “Lexical Chains as Representation of Context for the Detection and Correction of Malapropism.” In Fellbaum, C. (ed) WordNet: An Electronic Lexical Database and Some of its Applications. MIT Press, Cambridge, Mass.

Hobbs, J., Stickel, M., Appelt, D., and Martin, P., (1993) "Interpretation as Abduction." Artificial Intelligence 63, 69--142.

HotBot, http://www.hotbot.com

Hsu, Wen-Lin and Lang, Sheau-Dong (1999) "Classification Algorithms for NETNEWS Articles" Proceedings of CIKM-99, 8th ACM International Conference on Information and Knowledge Management.

Ide, N. and Vėronis, J. (1998) "Word Sense Disambiguation: The State of the Art" Computational Linguistics, 24:1, 1-40. Läses översiktligt.

ISO: International Organization for Standards, http://www.ISO.org

Jain, A. K., Murty,M. N., and Flynn, P. J. (1999) "Data Clustering: A Review." ACM Computing Surveys, vol. 31, No. 3, September, pp.255-323.

Jin, Rong, Christos, Falusos, and Hauptmann, G. Alex (2001) "Meta-Scoring: Automatically Evaluating Term Weighting Schemes in IR without Precision-Recall." In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, United States.

Joachims, Thorsten. (1997) "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization." In International Conference on Machine Learning (ICML).

Joachims, Thorsten. (1998) "Text Categorization with Support Vector Machines: Learning with Many Relevant Features." In European Conference on Machine Learning (ECML), Berlin, pp. 137-142.

Kehagias, A., Petridis,V., Kaburlasos,V.G., and Fragkou, P. (2001) "A Comparison of Word- and Sense-Based Text Categorization Using Several Classification Algorithms." Journal of Intelligent Information Systems, vol. 21, Issue 3.

Klavans, Judith; Chodorow, Martin, and Wacholder, Nina (1990) "From Dictionary to Knowledge Base via Taxonomy." Proceedings of the 6th Conference of the UW Centre for the New OED, Waterloo, Canada, 110-132.

Kleinberg, J. M. (1999) "Authoritative Sources in a Hyperlinked Environment." Journal of the ACM, vol. 46, Nr. 5, pp. 604-632.

Koller, D. and Sahami, M. (1998) "Hierarchically Classifying Documents Using Very Few Words." Proceedings of the 14th International Conference on Machine Learning ECML98.

Labrou, Yannis and Finin, Tim. (1999) "Yahoo as an Ontology – Using Yahoo Categories to Describe Document." In CIKM '99. Proceedings of the Eighth International Conference on Knowledge and Information Management, 180-187, ACM.

Lang, K. (1995) "Newsweeder: Learning to Filter News." In Proceedings of the 12th International Conference on Machine Learning, 331-339.

Larsen, B. and Aone, C. (1999). "Fast and Effective Text Mining Using Linear-Time Document Clustering." In Proc. of the Fifth ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining, pp. 16-22.

Lertnattee, V., and Theeramunkong, T. (2001) "Improving Centroid-Based Text Classification Using Term-Distribution-Based Weighting and Feature Selection." In Proceedings of INTECH-01, 2nd International Conference on Intelligent Technologies, Bangkok, Thailand, 2001, pp. 349-355.

Levow, G. A. (1997) "Corpus-based Techniques for Word Sense Disambiguation." Technical Report AIM-1637, MIT AI Lab, 1.

Lewis, D.D. and Jones, K.S. (1996) "Natural Language Processing for Information Retrieval," Comm. of the ACM, vol.39, pp.92-101.

Lextek stop list: http://www.lextek.com/manuals/onix/stopwords1.html

Li, Xiaobin, Matwin, Stan, and Szpakowicz, Stan (1995) "A WordNet-based Algorithm for Word Sense Disambiguation." Proceedings of IJCAI-95. Montréal, Canada.

Luk, A. (1995) "Statistical Sense Disambiguation with Relatively Small Corpora Using Dictionary Definitions." In Proceedings of the 33rd Meetings of the Association for Computational Linguistics (ACL-95), pages 181-188, Cambridge, M.A.

Lycos, http://www.lycos.com

Marcu, Daniel. (1997) "From Discourse Structures to Text Summaries." Proceedings of the ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization.

Microsoft: http:// www.Microsoft.com

Miller, G. A., Beckwith R., Felbaum C., Gross D., and Miller K., (1990) "Introduction to WordNet : An On-line Lexical Database." International Journal of Lexicography, 3, (4), 235 - 244. Report No. 43, Princeton University.

Miller, George A. (1990) "Nouns in WordNet: a Lexical Inheritance System." In International Journal of Lexicography 3 (4).

Mladenic, Dunja (1998) “Machine Learning on Non-Homogeneous, Distributed Text Data.” PhD thesis, University of Ljubljana, Slovenia.

Page, L., Brin, S., Motwani, R., and Terry Winograd, T. (1998) “The Page Rank Citation Ranking: Bringing Order to the Web.” Technical Report, Stanford, (Santa Barbara, CA 93106, January 1998)

Paice, C. D. (1990) “Constructing Literature Abstracts by Computer: Techniques and Prospects.” In Information Processing and Management, 26(1), 171-186.

Pantelm, Patrick and Lin, Dekang. (2002) “Discovering Word Senses from Text.” Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining-2002.

Peng, X. and Choi, B. (2002) “Automatic Web Page Classification in a Dynamic and Hierarchical Way.” IEEE International Conference on Data Mining, pp. 386-393.

Peng, Xiaogang. (2004) “Sense-Based Classification on Text Documents by Semantic Hierarchy Representation” Dissertation, Louisiana Tech University. Expectied Nov. 2004.

Pierre, John M. (2000) “Practical Issues for Automated Categorization of Web Sites” http://64.233.179.104/search?q=cache:0jQK-iKbbq4J:www.sukidog.com/jpierre/ECDL2000.pdf+%22John+M.+Pierre%22+metaandhl=zh-CN

Porter, M.F. (1980) “An Algorithm for Suffix Stripping,” Program, 14(3) :130-137. It has since been reprinted in Sparck-Jones, Karen, and Peter Willet, 1997, Readings in Information Retrieval, San Francisco: Morgan Kaufmann, ISBN 1-55860-454-4.

Prasad, Ard. (1999) “Chaos! The Name is Internet.” DRTC Workshop on Information Management. Jan.

Quillian, M. Ross. (1961) “A Design for an Understanding Machine.” Communication presented at the colloquium Semantic Problems in Natural Language. September. King’s College, Cambridge University, Cambridge, United Kingdom.

Rau L. F., R. Brandow, and K. Mitze. (1994) “Domain-Independent Summarization of News.” In Dagstuhl Seminar Report 79: Summarising Text for Intelligent Communication, B. Endres-Niggemeyer, J. Hobbs, and K. Sparck-Jones, editors, Dagstuhl, Germany.

Reuters 21578 http://about.reuters.com/researchandstandards/corpus/

Resnik, P. (1995). “Using Information Content to evaluate Semantic Similarity in a Taxonomy.” In Proceedings of IJCAI.

RFC 1738: Uniform Resource Locators (URL) http://www.faqs.org/rfcs/rfc1738.html

Rodriguez, Buenaga M., Gmez-Hidalgo, J. M., and Daz Agudo, B. (1997) "Using WordNet to Complement Training Information in Text Categorization." In Proceedings of the International Conference on Recent Advances in Natural Language Processing," Tzigov Chark.

Sahami, M., Hearst, M., and Saund, E. (1996) "Applying the Multiple Cause Mixture Model to Text Categorization." In Proceedings of the 13th International Conference in Machine Learning, pp. 435-443.

Salton, G. and Buckley, C. (1988) "Term Weighting Approaches in Automatic Text Retrieval." Information Processing and Management, 24, pp. 513-523.

Scott, S. and Matwin, S. (1998) "Text Classification Using WordNet Hypernyms." In Harabagiu, S. ed.: Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference, Somerset, New Jersey, Association for Computational Linguistics. pp. 38-44.

Sebastiani, F. (2002) "Machine Learning in Automated Text Categorization." ACM Computing Surveys 34(1), pp. 1-47.

Steiner, E., Eckert, U., Roth, B., and Winter, J. (1988). "The Development of the EUROTRA-D System of Semantic Relations," In: Steiner, E., Schmidt, P., and Zelinsky-Wibbelt (eds), from Syntax to Semantics: Insights from Machine Translation, London: Frances Printer.

Sun, A., Lim, E., and Ng, W. (2002). "Web Classification Using Support Vector Machine." WIDM '02, McLean, Virginia, USA.

USA Today, (2002) "Search Engines not Differentiating Ads from Results" July 12, http://www.usatoday.com/tech/news/2002/07/12/online-search-engines.htm

Vapnik, V. (1995) "The Nature of Statistical Learning Theory." Springer, New York.

Vossen, P., Diez-Orzas, P., and Peters, W. (1997) "The Multilingual Design of EuroWordNet." In P. Vossen, N. Calzolari, G. Adriaens, A. Sanfilippo, Y. Wilks (eds.) Proceedings of the ACL/EACL-97 Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, Madrid, July 12th.

Waltz, David L. and Pollack, Jordan B. (1985) "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation." Cognitive Science, 9, 51-74.

Weiss, D. (2001) "A Clustering Interface for Web Search Results in Polish and English." Master's Thesis, Poznan University of Technology, Poland, June.

Weitzner, Daniel (2003) "Semantic Web: Creating the Web Effect for Data." Presented at INTAP Interoperability Technology Association for Information Processing, November 11, 2003 Tokyo, Japan

Wnstats - WordNet 2.0 database statistics: http://www.cogsci.princeton.edu/~wn/man/wnstats.7WN.html

WordNet: http://www.cogsci.princeton.edu/~wn/

WordNet Search 2.0, http://www.cogsci.princeton.edu/cgi-bin/webwn

Yahoo: http://www.Yahoo.com

Yang, Y. and Liu, X. (1999) "A Re-Examination of Text Categorization Methods." In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, US.

Yang, Y. and Pederson, J.O. (1997) "A Comparative Study on Feature Selection in Text Categorization." Proceedings of the Fourteenth International Conference on Machine Learning pp. 412 - 420. ISBN:1-55860-486-3.

Yang, Y., Slattery, S., and Ghani, R. (2001) "A Study of Approaches to Hypertext Categorization." Journal of Intelligent Information Systems, Special Issue on Automatic Text Categorization.

Yarowsky, David. (1992). "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora." In Proceedings of the 14th International Conference on Computational Linguistics (COLING-92), pp. 454-460, Nantes, France.

Yarowsky, David. (1995) "Unsupervised Word-Sense Disambiguation Rivaling Supervised Methods." In Proceedings of the 33rd Annual Meeting of the Association for Computational Lainguistics (ACL '95), pp. 189-196, Cambridge, MA.

Zaiane, O. R. and Antonie, M. L. (2002) "Classifying Text Documents by Associating Terms with Text Categories." In Proceedings of the 13th Australasian Database Conference (ADC '02), Melbourne, Australia.

Zamir, O. and Etzioni, O. (1998) "Web Document Clustering: A Feasibility Demonstration." in Proc. 21st Annu. Int. ACM SIGIR Conf., pp.46–54.

Zelinsky-Wibbelt, C. (1988) "From Cognitive Grammar to the Generation of Semantic Interpretation in Machine Translation." In Steiner, E., Schmidt, P., and Zelinsky-Wibbelt (eds), From Syntax to Semantics: Insights from Machine Translation, London: Frances Printer.

Zhang, T. and Oles, F. J. (2001) "Text Categorization Based on Regularized Linear Classification Methods." Information Retrieval, v.4 n.1, p.5-31, April.

Zhao, Y. and Karypis, G. (2002) "Evaluation of Hierarchical Clustering Algorithm for Document Datasets." In Proceedings of the Eleventh International Conference on Information and Knowledge Management. Web Clustering Session. pp.515-524 http://www.users.cs.umn.edu/~karypis/publications/datamining.html.