LIBRARY
Michigan State
University

This is to certify that the
dissertation entitled

INCORPORATING BACKGROUND KNOWLEDGE IN DOCUMENT CLUSTERING

presented by

SAMAH JAMAL FODEH

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Computer Science

Major Professor's Signature

8/19/2010

Date

MSU is an Affirmative Action/Equal Opportunity Employer

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

5/08 K:/Proj/Acc&Pres/CIRC/DateDue.indd

INCORPORATING BACKGROUND KNOWLEDGE IN DOCUMENT CLUSTERING

By

SAMAH JAMAL FODEH

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science

2010

ABSTRACT

INCORPORATING BACKGROUND KNOWLEDGE IN DOCUMENT CLUSTERING
By

SAMAH JAMAL FODEH

The explosive growth of unstructured text data in the present digital age has triggered an overwhelming interest in the development of robust and scalable document clustering techniques that can automatically partition and summarize the large tracts of documents. As document clustering is an unsupervised learning task, the quality of the partitions may be suboptimal due to the lack of guidance about which documents belong together in the same cluster. Augmenting the clustering algorithm with additional side information may potentially lead to better clusters. Towards this end, this thesis focuses on the use of background knowledge from an ontology such as WordNet to enhance the performance of document clustering algorithms. There are numerous challenges that must be overcome in order for such an approach to be successful. Most notably, how to effectively map the original words in the documents to their corresponding concepts in an ontology? The strategy used for concept mapping is important because it may increase the dimensionality of the data or introduce erroneous concepts, both of which have an adverse effect on the quality of the final partitions. In addition, the choice of ontology is another factor that should be taken into consideration since each ontology has its own structure, coverage, and content. Despite these challenges, a considerable amount of research has been done over the past decade on ontology-driven clustering. Yet the results from previous studies have not been conclusive. Some concluded that ontology helps improve clustering performance while others showed it is not that helpful. This

thesis investigates the various factors that affect the performance of such clustering algorithms, including the choice of ontology, concept mapping approach, and benchmark datasets and baseline algorithms used for evaluation. The contributions of this thesis are as follows: First, a noun-based approach is proposed as a simple but more stringent baseline for clustering. Second, a novel unsupervised information gain approach is developed for extracting a core subset of semantic features from an ontology that can be effectively used for clustering. Third, a hybrid ontology-driven ensemble clustering method is proposed that combines the clusters of nouns and clusters of concepts extracted from an ontology. Finally, an approach for extracting concepts from Wikipedia is proposed and compared against existing works. These concepts are then used in conjunction with the concepts (synsets) from WordNet to study the effect of applying multiple ontologies on document clustering.

ACKNOWLEDGEMENTS

In the name of ALLAH, most gracious most merciful; All the praises and thanks be to Allah. I am grateful to ALLAH who permits me to live and allowed me to start and finish this journey.

My first and most important thanks should go to my advisors Dr. Pang-Ning Tan and Dr. William Punch for their guidance, inspiring thoughts and patience throughout my graduate study at MSU. Their endless support and mentorship made this thesis possible. Special thanks go to Dr. Joyce Chai and Dr. Vincent Melfi for agreeing to serve on my committee.

I would like to thank the department of Computer Science and Engineering for giving me the opportunity to pursue my Ph.D. In particular, many thanks go to Dr. Laura Dillon and Dr. George Stockman for being there for me whenever I was in need. I wish to thank all my fellows in the data mining group, especially Dr. Kapila who passed away last year of heart attack.

Most earnest thanks go to my father Jamal and my mother Horeyeh Hamdan for their faith in me. It is under their watchful eye that I finished my pursuit. My Most heartfelt acknowledgement goes to my Husband Fuad, his unwavering love and companionship gave me the self-steam to continue this journey. I also owe a huge gratitude to my kids, Haya, Khaled and Amir. My wonderful kids, please forgive me for all the moments I had to work away from you. We will make it up in the days to come. At last, I would like to thank my brothers and sisters for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES viii

LIST OF FIGURES x

Chapter 1
Introduction 1
1.1 Motivation 1
1.2 Thesis Overview 4
1.3 Contributions 8
1.3.1 Development of a Simplified Noun-based Approach for Baseline Document Clustering (Chapter 4) 8
1.3.2 Concept Selection using Information Gain (Chapter 4) 9
1.3.3 Clustering Documents using WordNet (Chapter 5) 10
1.3.4 Clustering Documents using Wikipedia (Chapter 6) 10
1.4 Outline of the Thesis 11

Chapter 2
Related Work 12
2.1 Document Clustering 12
2.2 Semi-Supervised Clustering 14
2.3 Ontology-based Clustering 14
2.3.1 Ontology Structure 15
2.3.1.1 WordNet 15
2.3.1.2 Wikipedia 18
2.3.1.3 MeSH 19
2.3.2 Clustering using Ontology 20
2.3.2.1 WordNet 20
2.3.2.2 Wikipedia 25
2.3.2.3 MeSH 28
2.4 Summary 30

Chapter 3
Data and Preprocessing 31
3.1 Data 31
3.1.1 Reuters-21578 31
3.1.2 20Newsgroups 32
3.2 Preprocessing 35
3.3 WordNet-Based Concept Mapping 37
3.4 Wikipedia-Based Concept Mapping 41

3.4.1 Preprocessing Wikipedia Articles 41
3.4.2 Document Concept-based representation 43
3.5 Summary 47

Chapter 4
Effect of Preprocessing on Document Clustering 48
4.1 Evaluation Metrics 48
4.2 Baseline Clustering Algorithm 50
4.3 Effect of Preprocessing 50
4.3.1 Why Nouns? 53
4.3.1.1 Nouns and Centroids Analysis 54
4.3.1.2 Nouns and LSI Analysis 56
4.3.1.3 Nouns versus Random Words: Comparison 58
4.4 Feature Selection using Information Gain 60
4.4.1 Information Gain 60
4.4.2 Core Semantic Feature Selection with Information Gain 63
4.4.3 Experimental Evaluation 65
4.4.3.1 Clustering using Core Semantic Features 65
4.4.3.2 Clusters Refinement 70
4.4.3.3 Effect of using Core Semantic Features 72
4.5 Summary 76

Chapter 5
Clustering Documents using WordNet: An Ensemble Approach 77
5.1 Preliminaries 77
5.2 Proposed Framework 78
5.3 Noun Cluster Membership Matrix 79
5.4 Sense Cluster Membership Matrix 80
5.5 Combined Ensemble Clustering 81
5.6 Experiments 83
5.6.1 Document Clustering using Senses 83
5.6.2 Ensemble Clustering 85
5.6.3 Value of Sense Information 91
5.7 Summary 94

Chapter 6
Clustering Documents using Wikipedia: Concept Mapping Approach 95
6.1 Exploiting Wikipedia 95
6.2 Evaluation of Wikipedia-based Concept Mapping Approach 96
6.3 Exploiting Categories 100
6.4 Combining Multiple Sources of Knowledge 106
6.5 Summary 111

Chapter 7
Conclusion and Future Directions 113
7.1 Contributions of this Thesis 114

7.1.1 Development of a Simplified Noun-based App. for Baseline Clustering .. 114
7.1.2 Clustering using Core Features with High Information Gain 114
7.1.3 Hybrid Ensemble Approach for Document Clustering 115
7.1.4 Exploit Wikipedia for Document Clustering and Combine Multiple Sources of knowledge 115
7.2 Future Directions 116

REFERENCES 124

LIST OF TABLES

Table 1: Document titles from Reuters-21578 news dataset 3

Table 2: Document titles from Technorati.com 4

Table 3: Nouns lexical categories in WordNet 23

Table 4: Results of incorporating ontologies in document clustering. Ont. is short of Ontology 24

Table 5: 20Newsgroups Classes 33

Table 6: Number of different features in Binary-Multiple-500 dataset 34

Table 7: List of some of the words that appear in the document *d* 46

Table 8: Top concepts mapped to the document *d* using Gabrilovich approach 46

Table 9: Top concepts mapped to the document *d* using our proposed approach 47

Table 10: Comparison between the nouns baseline (column 5) against reported baseline (column 4) and reported clustering methods in the literature (column 3) in terms of purity. (*) This result was created using 60 clusters to make an equivalent comparison. (**) This result was created using 4 clusters 52

Table 11: Fraction of nouns among the top 20 words of each cluster centroid 53

Table 12: Comparison between the noun-based and word-based centroids 54

Table 13: Percentage of nouns in different percentiles in the top 20 components 57

Table 14: Percentage of nouns in top k words with highest mutual information 58

Table 15: Document distribution across clusters before and after disambiguation 62

Table 16: Comparison between the core semantic feature (CSF) approach against three baseline methods (using all nouns, all concepts or Hotho's method) in terms of cluster purity and amount of feature reduction. (see discussion on how we may imve the purity of the multi-class datasets with the *). N(Nouns), C(concepts) 68

Table 17: Threshold-based centroid mapping 72

Table 18: Top 15 features of the centroids in the clusters using nouns and the clusters using the core concepts 75

Table 19: Comparison of entropy for K-means 84

Table 20: Ensemble results for nouns senses and the both combined for Binary-Multiple-500 90

Table 21: Average WMW statistics 93

Table 22: Comparison between Wikipedia concepts, T_{depth4}, and T_{3Up} clusters in terms of Purity 97

Table 23: Comparison with other methods using min20-max200 Reuters dataset 98

Table 24: Comparing features generated by different approaches 99

Table 25: Performance of the ensembles in terms of entropy 108

Table 26: Performance of the ensembles in terms of Purity 108

Table 27: Performance of the ensembles in terms of NMI 109

Table 28: List of top 46 Topics in Reuters 118

Table 29: List of Stopwords 120

LIST OF FIGURES

Figure 1: A snapshot of the Yippy clustering search engine 2

Figure 2: A snapshot of the ClusterMed search engine for retrieving abstracts of biomedical research articles from the PubMed digital library 2

Figure 3: Concepts of the noun "currency" in WordNet 16

Figure 4: Category-subcatagory structure in Wikipedia 17

Figure 5: The disambiguation page of "currency" 18

Figure 6: Part of MeSH tree 19

Figure 7: Flowchart of the reprocessing steps applied to the original documents 37

Figure 8: semantic network between the first senses of dog and cat 39

Figure 9: Relatedness score computations via matrix multiplication 44

Figure 10: Purity of clusters of nouns versus random words 59

Figure 11: Distances of documents from centroids of clusters using the core concepts .. 73

Figure 12: Distances of documents from centroids of clusters using all the concepts 74

Figure 13: Comparison of entropy values of Ens_{noun}, Ens_{sense} and Ens_{both} clusters for Multiple6000 and Reuters2 datasets 86

Figure 14: Comparison of purity values of Ens_{noun}, Ens_{sense} and Ens_{both} clusters for Multiple-6000 and Reuters2 datasets 88

Figure 15: Comparison between k-means using cosine and the combined ensemble 89

Figure 16: Ensemble results for nouns senses and both combined for Binary-M.-500 91

Figure 17: Part of the category structure in Wikipedia, rectangles denote concepts while circles denote categories 101

Figure 18: Cumulative distribution function of Wikipedia T_{depth4} categories and nouns for Reuters2 dataset 105

Figure 19: Comparison of the different ensembles with the Baseline in terms of Purity 110

Figure 20: Comparison of the diff. ensembles with the Baseline in terms of Entropy... 110

Figure 21: Comparison of the different ensembles with the Baseline in terms of NMI. 111

Chapter 1

Introduction

Document clustering seeks to partition a set of documents into groups containing similar topics. It has been an active area of study in the last fifty years, spurred by the explosive growth of large tracts of unstructured text data on the World Wide Web as well as in public, commercial, and medical databases. Techniques for document clustering have been deployed in many information retrieval and text mining systems, including Yippy, Cuil, WebClust, Carrot, and Vivisimo. As an example, *Yippy* (formerly known as Clusty) is a general-purpose search engine that summarizes the retrieval results of queries entered by users in a clustered fashion, as shown in Figure 1. ClusterMed is another clustering-based search engine designed for accessing biomedical abstracts from the PubMed digital library, while Lexis Nexis provides capabilities for clustering the search results of legal documents. A snapshot of the ClusterMed user interface is shown in Figure 2.

1.1 Motivation

The advantage of using document clustering is its ability to find natural groupings of documents in an unsupervised fashion, thus allowing it to be applied to any text corpus with minimal human intervention. However, the flexibility afforded by clustering is also its pitfall, since individual datasets have their own characteristics, making it difficult to design a clustering methodology that works effectively on all datasets.

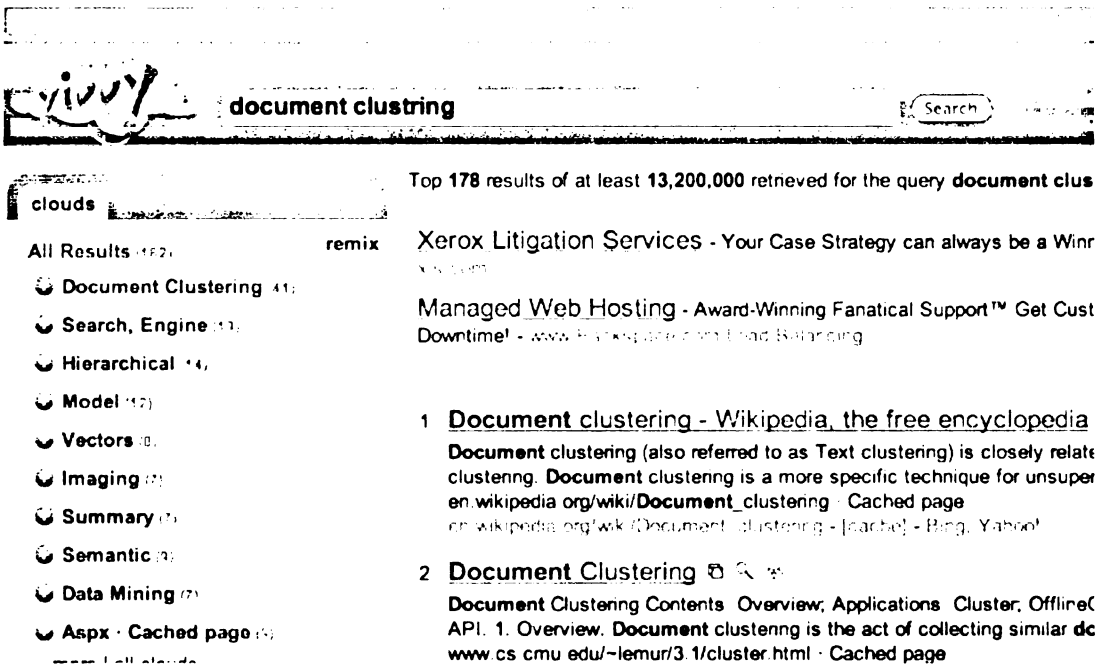


Figure 1: A snapshot of the Yippy clustering search engine

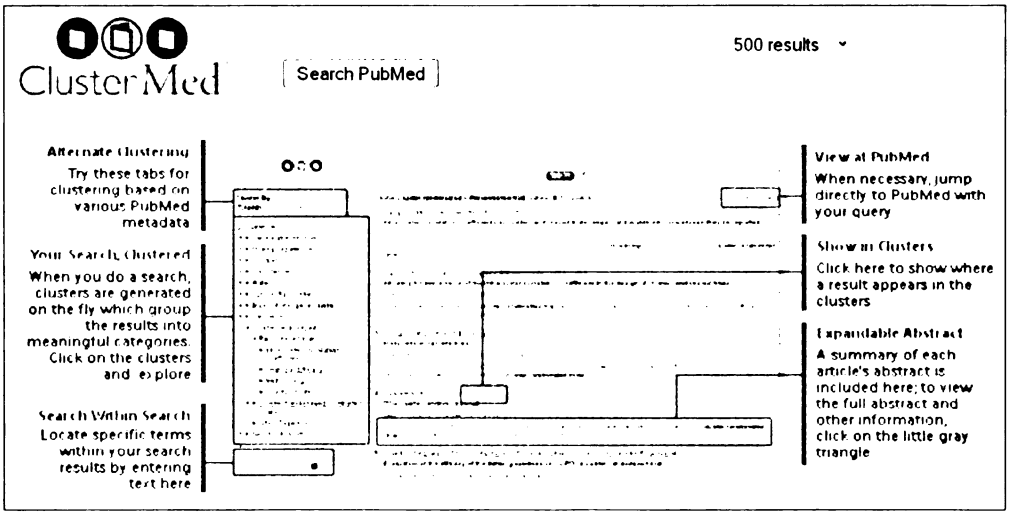


Figure 2: A snapshot of the ClusterMed search engine for retrieving abstracts of biomedical research articles from the PubMed digital library

To illustrate the challenges in document clustering, consider the sample dataset shown in Table 1, which contains the titles of 3 news stories from the Reuters-21578 dataset. The first two titles belong to the "grain" category while the last title belongs to the "sugar" category. Without any additional information, it is hard to distinguish the first two

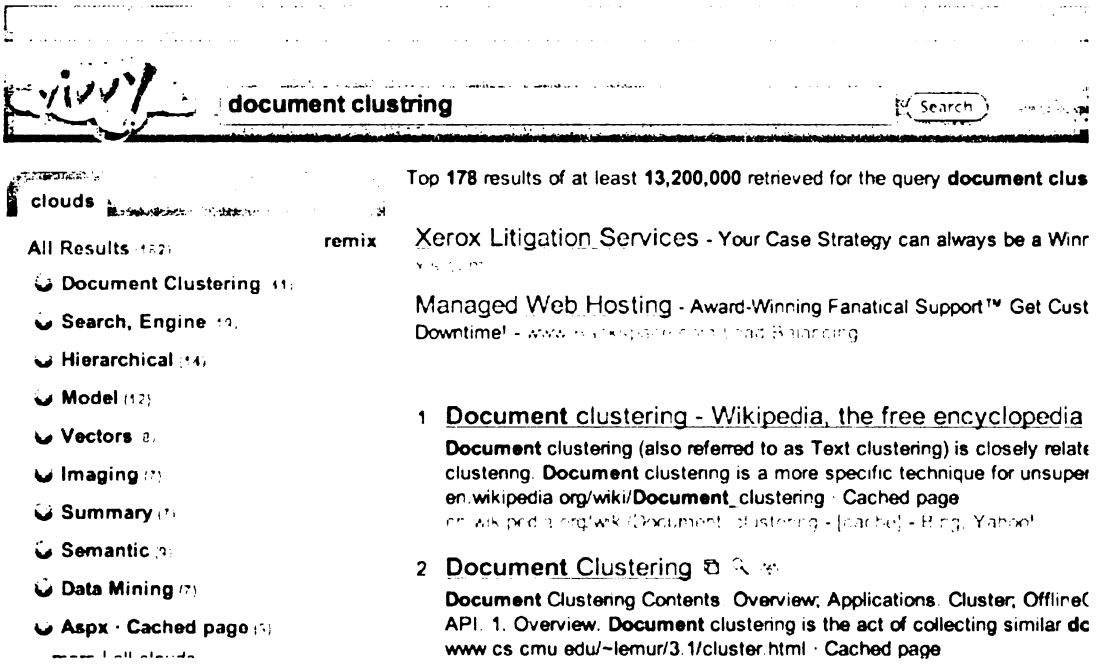


Figure 1: A snapshot of the Yippy clustering search engine

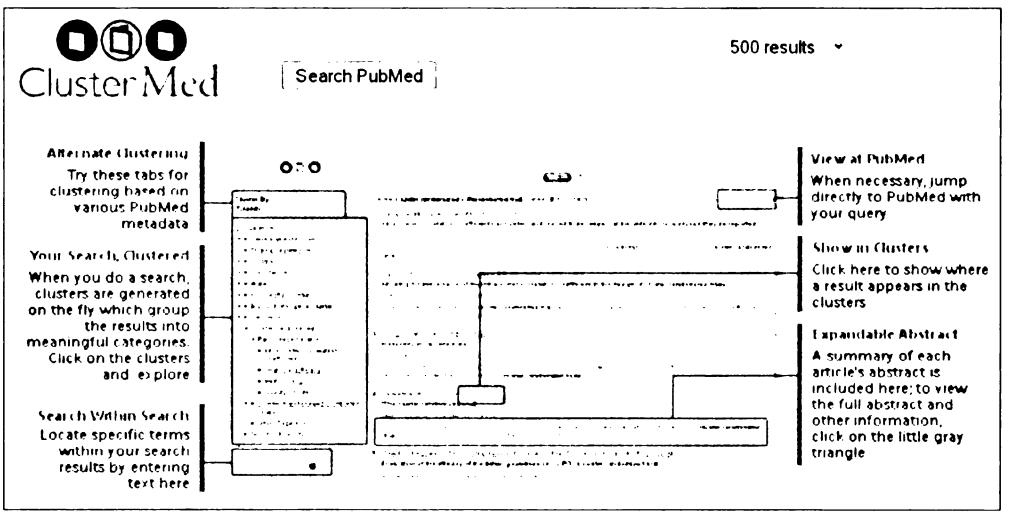


Figure 2: A snapshot of the ClusterMed search engine for retrieving abstracts of biomedical research articles from the PubMed digital library

To illustrate the challenges in document clustering, consider the sample dataset shown in Table 1, which contains the titles of 3 news stories from the Reuters-21578 dataset. The first two titles belong to the "grain" category while the last title belongs to the "sugar" category. Without any additional information, it is hard to distinguish the first two

documents from the last one due to the presence of their common word "USDA". One way to overcome this problem is by incorporating *instance-level constraints* such as those provided in semi supervised clustering [7][15][111] where must-link (ML) and cannot-link (CL) constraints are used to determine which two objects should be placed in the same cluster [67][111]. This explicit information suggests the availability of an expert in the domain who would annotate the labels of documents or summarize the important associations between documents in the dataset. In practice, this human intervention can be expensive and could produce inaccurate results depending on the reliability of the information provided.

Doc#	Topic	Title
1	Grain	USDA estimates Soviet wheat, coarse grains
2	Grain	USDA said unlikely to broaden corn bonus offer
3	Sugar	U.S. sugar imports down in week - USDA

Table 1: Document titles from Reuters-21578 news dataset

As another example from the online social media domain, consider the titles of four blog postings[1] shown in Table 2. The first two titles are from the computer security topic while the last two titles are related to medicine. Existing algorithms will have difficulty clustering the blog postings correctly for several reasons. The first and third titles are likely to be assigned to the same cluster even though they belong to different topics. This is because they both contain the same word "virus", which has a different meaning in each document. This confusion can be avoided if the semantic meaning of the polysemous word "virus" in the different titles is resolved (e.g., by applying WordNet).

[1] The titles of the blog postings are obtained from technorati.com.

Second, the third and fourth titles do not share any words in common even though they belong to the same topic. Their similarity is zero which decreases the likelihood they will be put in the same cluster. Nevertheless, the titles contain the synonyms "swine flu" and "H1N1". With the help of an ontology such as WordNet or Wikipedia, the semantic relation between these words can be established in order to increase the similarity between the two blog postings. This example motivates the use of ontology as side information to guide the clustering algorithm into finding better clusters [35][92].

Doc#	Topic	Title
1	Computer Security	**Virus** Hits Computer Network of Swiss Foreign Ministry
2	Computer Security	Swine flu and the computer virus
3	Medicine	*Swine flu* 'related' to 1918 pandemic **virus** - survivors exhibit resistance
4	Medicine	Special *H1N1* vaccine for pregnant women is available

Table 2: Document titles from Technorati.com

1.2 Thesis Overview

The goal of this thesis is to investigate the value of incorporating background knowledge from a given ontology into document clustering. Following the definition by Hotho et al. [35], an ontology O is a poset $(C, \prec_C)$ consisting of a set of concept identifiers C and a partial order $\prec_C$ imposed on C. The partial order can be represented by a directed, acyclic graph known as a *concept hierarchy*. Concepts are nodes in the hierarchy, which are also called synsets or descriptors (depending on the type of ontology).

WordNet is a general-purpose ontology that can be used to generate a semantic representation of documents. This new representation is obtained by mapping the original

words in the documents to their best evaluated senses (synsets) using the WordNet hierarchy. This process is known as Word Sense Disambiguation (WSD). If performed correctly, WSD may alleviate two fundamental problems in text clustering: *polysemy* and *synonymy*. Polysemy corresponds to terms that have multiple meanings, depending on the context in which they are used. For example "bank" is a polysemous term that may refer to a "river bank" or a "financial institution" depending on the content of the document. Synonymy refers to the case where a group of words serve the same meaning. For example the words "different", "diverse", "dissimilar" express the same meaning. As shown in the examples of Section 1.1, polysemy and synonymy have a direct effect on increasing or decreasing the similarity between documents, which in turn affect the results of clustering. With the help of an ontology, these words can be disambiguated and the appropriate synsets they represent are used to cluster the documents. In addition, WordNet can be used to introduce more general concepts (hypernyms) of the current words in the documents. The use of hypernyms helps to increase the similarity of related documents that do not share any words in common. For example, consider a pair of sports-related documents; one document is about "tennis" while the other is about "soccer". Since "tennis" and "soccer" are both descendents of the synsets "game" and "sport" in the WordNet hierarchy, augmenting the synsets "game" and "sport" into both documents would help increase their similarity. Nevertheless, as will be shown in this thesis, the usefulness of using WordNet as an ontology for clustering is subject to the correctness of using the WSD approach to resolve these terms.

Likewise, Wikipedia can be used to create a new representation of documents, in which each concept corresponds to a Wikipedia article. The relation between concepts

can be established based on the high-level categories defined by Wikipedia editors. For example, "Cluster Analysis" and "Association rule learning" are two concepts in Wikipedia that belong to "Category: Data Mining", which in turn, is part of "Category: Data Analysis". It is important to mention that Wikipedia is not exactly an ontology because its category network does not form a directed acyclic graph. Nevertheless, it is possible to derive a Wikipedia ontology by removing the cycles in the network.

The background knowledge provided by an ontology to clustering differs from that provided by semi-supervised learning. While semi-supervised clustering provides specific information about the data set (what documents go where), an ontology oriented approach provides general information about the relationship between terms that appear in a document and is often applicable to any text corpus. Thus, background knowledge from an ontology can be regarded as a form of *feature-level constraints* whereas semi-supervised learning imposes *instance-level constraints* to the clustering algorithm.

Despite its advantages, there are many challenges in incorporating an ontology into document clustering. First and foremost, ***how to effectively map the words in a document to their corresponding concepts in an ontology?*** Typically, there are many possible concepts that can be mapped to a given word in a document. Selecting the appropriate concepts to represent the words in a document can have a significant impact on clustering. ***Should we use the context of the word to select the most appropriate concept or is it possible to include all the concepts?*** The former would require specifying a window size (e.g., at sentence-level, paragraph-level, or document-level) to define the context of a word. The latter approach, on the other hand, may increase the dimensionality of the data and possibly introduce erroneous features that degrade

clustering performance. The increasing dimensionality problem may also arise even if each word is replaced by its most appropriate direct concept. This is because one may have to include ancestors of the concepts (e.g., hypernyms of synsets in WordNet hierarchy) in order to increase the similarity of two related documents that have no common words. The challenge of using the ancestor nodes is ***how to determine up to which level in the concept hierarchy to use?*** Finally, since most concept mapping approaches are imperfect, the challenge is ***how to handle noisy features introduced by the mapping process?***

In addition to the previous challenges, the characteristics on an ontology also play a significant role in the quality of concepts extracted. In particular, the scope and content of an ontology affect completeness of the extracted features. For instance, WordNet has a limited coverage because it was built manually by a small group of experts. Furthermore, the 'gloss' (a small text description of a term in the taxonomy in WordNet) is not sufficient to explain the contextual uses of a term. In contrast, Wikipedia has a broader coverage and richer content since each concept is associated with a unique article that often contains long paragraphs of text describing the concept and its alternate usage. Given their intrinsic differences, an obvious research question is, "***Does combining multiple ontologies help reconcile the deficiencies in individual ones?***"

Research on incorporating ontology into document clustering has grown over the past decade in spite of these challenges. So far, the results from prior studies have not been conclusive. While some researchers concluded that incorporating background knowledge from ontology helps to improve clustering, others have reached an opposite conclusion. We argue that the contradictory findings are caused by several factors,

including the benchmark data and baseline algorithm used to evaluate the proposed ontology-driven clustering methods (see Table 4 in Chapter 2). Some researchers have used a much weaker baseline to conclude ontology helps while others used a more rigorous baseline and did not find any significant improvements. The challenges arising from incorporating background knowledge from WordNet or Wikipedia into document clustering suggest that more research will be needed to effectively utilize the background knowledge.

1.3 Contributions

This thesis aims to investigate the added value of using ontology into document clustering. In addition to proposing new methods for concept mapping, feature selection, and clustering, we also analyze characteristics of the data to help provide a better understanding of why some methods are successful or fail when applied to certain datasets. The contributions of this thesis are summarized below

1.3.1 Development of a Simplified Noun-based Approach for Baseline Document Clustering (Chapter 4)

To demonstrate the effectiveness of any new clustering algorithm, one has to compare its performance against some baseline approaches on the same datasets. Unfortunately, researchers in document clustering have been using different baseline approaches to generate their clusters, some of which are much weaker than others. In our opinion, a baseline clustering approach should reflect the best clusters that can be obtained from the data with a relatively cheap clustering algorithm. In this thesis, we show that a relatively simple baseline approach using spherical kmeans and the document's ``nouns'' as the features produces significantly better baseline clusters, often times competitive with the more complicated methods. Our approach is simple because we do not strictly enforce the

identification of a noun using part of speech (PoS) approaches. Rather, we use the WordNet-based Morphy stemmer to both stem the word and determine if the word could be a noun without strictly observing its PoS in the sentence. That is, if the word can act as a noun, we treat it as such. Results are provided that compare typical baseline results from published research, as well as the associated improved method results, to our noun-based approach. Our hope is that this baseline can be used by other researchers as a higher standard for their algorithm development.

1.3.2 Concept Selection using Information Gain (Chapter 4)

A document may contain irrelevant and redundant words that are not related to the topic of the document. The presence of such words may distort the clustering results. Furthermore the concept mapping approach used to map the words to ontological concepts may introduce inaccurate semantic features and increase the dimensionality of the feature space. To address these problems, this thesis presents an information theoretic approach for extracting a core subset of semantic features from a given ontology that can be effectively used for clustering. Ideally, this subset of features should correspond to the main topics of the documents. Empirical results show that by using core semantic features for clustering, one can reduce the number of features by 80% or more and still produce clusters comparable to, and occasionally better than, those clusters produced using either all the document words or all the derived semantic features as the feature set.

1.3.3 Clustering Documents using WordNet (Chapter 5)

Ensemble clustering is the process of aggregating several clustering solutions to capture their consensus in terms of grouping similar documents in the same cluster. This thesis explores the use of ensemble clustering for combining the results of document clusters

obtained using the nouns in the documents and the document clusters obtained using the senses extracted from WordNet. This approach not only yields better clusters than the simplified nouns baseline but also performed slightly better than the ensemble of nouns and ensemble of senses that were constructed individually when applied to the majority of the benchmark datasets.

1.3.4 Clustering Documents using Wikipedia (Chapter 6)

This thesis presents an approach for extracting concepts from Wikipedia. We performed an experimental study to evaluate the approach and showed that it is more effective than existing concept mapping approaches. We also experimented with using Wikipedia categories as features. Finally, we investigated the problem of integrating Wikipedia with WordNet and nouns using a hybrid ensemble clustering framework. Unfortunately, our results showed that Wikipedia did not add any significant improvements to the final clusters (compared to using ensemble of WordNet and nouns). This might be related to the larger number of features (both relevant and irrelevant) introduced by Wikipedia during concept mapping compared to WordNet. Having a better set of concepts from Wikipedia is expected to lead to better clustering after the combination.

1.4 Outline of the Thesis

The organization of this thesis is as follows:

Chapter 2: Related Work.

This chapter gives a review of the recent research related to incorporating background knowledge into document clustering.

Chapter 3: Data and Preprocessing.

This chapter describes the preprocessing steps that are performed on the datasets. It also includes a detailed description of all the datasets that have been used in this thesis.

Chapter 4: Effect of Preprocessing on Document Clustering

This chapter describes the effect of preprocessing on document clustering. It also presents our proposed simplified noun approach for baseline clustering and a concept selection algorithm using information gain.

Chapter 5: Clustering Documents using WordNet

This chapter describes a hybrid ensemble approach that utilizes the nouns and the WordNet senses as features to improve document clustering.

Chapter 6: Clustering Documents using Wikipedia

This chapter describes a concept mapping approach to exploit Wikipedia concepts and categories in document clustering. It also includes an experimental study to evaluate the benefits of integrating Wikipedia concepts with WordNet senses and the nouns in a combined ensemble clustering approach.

Chapter 7: Conclusion and Future Directions

This chapter summarizes the thesis and includes discussion about possible paths of future work.

Chapter 2

2 Related Work

In this section we present a survey of using background knowledge in document clustering. Section 2.1 summarizes some document clustering methods. Semi-supervised clustering review is given in Section 2.2, and finally in Section 2.3 a detailed review of the Ontology-based clustering methods is given.

2.1 Document Clustering

Document clustering is the task of partitioning a dataset into groups containing similar documents. Several document clustering algorithms were developed such as k-means, Latent Semantic Indexing (LSI) [14], Probabilistic LSI [34], Latent Dirichlet allocation (LDA) [6] and Author-topic Models [88][98].

K-means is one of the most popular methods in clustering. It aims to partition the text corpus into k clusters where each document is assigned to the cluster with the nearest mean; several metrics can be used to compute the distance or similarity between the documents and the mean such as: Euclidean distance, city block and cosine similarity. K is an input parameter that should be provided to the algorithm. LSI [14] is another method used in document clustering. It assumes that there is an underlying semantic model in the text which is approximated using the terms' co-occurrences. The assumption is that terms appearing in the same context are more likely to have similar meanings. It is based on a technique called Singular Value Decomposition (SVD). Using SVD, a term-document matrix A is decomposed into three components as shown in equation (1):

$$A = USV' \quad (1)$$

where U and V` contain orthonormal vectors that correspond to the left and right singular vectors and S is a square diagonal matrix of the singular values. It is noticed that clustering the V matrix gives better results than clustering the original document-term matrix A. K-means and LSI are examples of clustering methods based on term frequency information in a dataset. Probabilistic methods were also developed for document clustering such as Probabilistic Latent Semantic Indexing PLSI, Latent Dirichlet allocation LDA and Author-topic models.

However due to the unsupervised nature of the task and the lack of information to determine the quality of the clusters, there have been considerable interest in incorporating side information or background knowledge to aid clustering. Currently, there are two general methods to incorporate background knowledge; *semi-supervised* clustering and o*ntology-based* clustering.

In *semi-supervised* clustering, few labeled examples are used as constraints to guide the clustering process. This approach proved to be effective in improving the learning accuracy. On the other hand, *ontology-based* algorithms are focused on incorporating knowledge available in ontologies into clustering. The difference between semi-supervised and ontological approaches is that the former acquires knowledge from domain experts which is often an expensive time consuming task whereas incorporating knowledge from ontologies is more flexible.

2.2 Semi-Supervised Clustering

Semi-supervised clustering algorithms [67][7][15][111][59] improve the quality of clusters by using labeled examples or other forms of user-provided constraints. For example, Wagstaff et. al. [111] and Bradley et al. [7] extended the standard k-means clustering algorithm to incorporate pairwise (must-link and cannot-link) constraints. A must link constraint specifies that two instances have to be in the same cluster whereas a cannot-link constraint indicate that two instances can not be placed in the same cluster. The main modification to classic k-means is to ensure that these constraints are not violated during cluster assignments. Lu et al. [67] developed an approach for handling pair-wise constraints using a probabilistic framework based on Gaussian mixture models. Their experiments on different datasets showed that the pairwise constraints can improve the performance of the clustering process. In addition to pair-wise constraints, a semi-supervised clustering approach that uses group-level and soft constraints has been proposed in [62]. The value of the soft constraints reflects the certainty of the prior knowledge that a pair of objects should or should not be in the same cluster.

2.3 Ontology-based Clustering

In general, an ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. In particular, a linguistic ontology is a taxonomy in which lexical entries that are semantically related are associated together in the hierarchy via a parent-child relationship. Ontologies are useful since they provide a rich source of semantic information that can be harnessed for document clustering. WordNet [45] and Wikipedia [43] are examples of linguistic ontologies used in document clustering. Another example is the MeSH ontology [41] which has been used to cluster biomedical abstracts in Pubmed digital repository. In the following sections we describe

the structure of these ontologies and we summarize some of the previous work on using them for clustering.

2.3.1 Ontology Structure

An ontology is a tuple $O := (C, \prec_C)$ consisting of a set of concept identifiers C, and a partial order $\prec_C$ on C [35]. Ontologies are known to be domain specific because they conceptualize the background knowledge of a specific domain. The tuple $(C, \prec_C)$ is called a concept hierarchy. Concepts are nodes in the hierarchy that represent a unit of meaning. Concepts are called senses, synsets or descriptors in other ontologies.

2.3.1.1 WordNet

WordNet [45] is a large lexical ontology of English, developed under the direction of George A. Miller. Lexical entries such as nouns, verbs and adjectives are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept (sense). Concepts are interlinked by means of semantic and lexical relations [45].

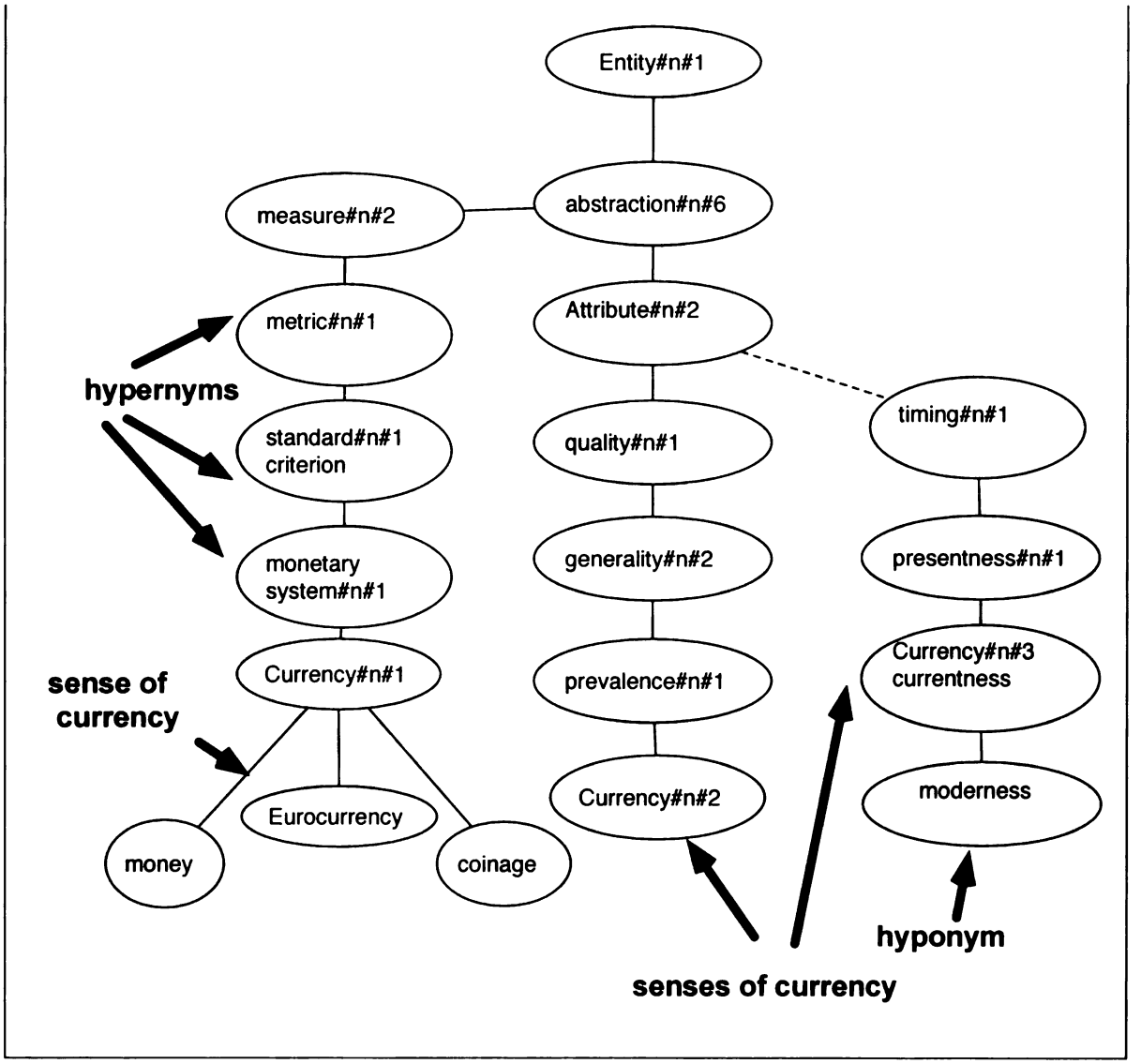


Figure 3: Concepts of the noun "currency" in WordNet

Figure 3 shows an example of the concepts of the noun "currency" in the hierarchy. Note that we did not include all the concepts for space considerations. As shown in the figure, "currency" has three senses: currency#n#1, currency#n#2 and currency#n#3; each sense has a unique ID. Several semantic relations exist in the WordNet hierarchy, some of which include:

- Hypernymy: is a semantic relation defined between a concept and its antecedent node in the hierarchy. For example, *monetary system* is a hypernym of currency#n#1 as appears from Figure 3.
- Hyponymy: is a semantic relation defined between a concept and its descendent node in the hierarchy. *Money* is a hyponym of currency#n#1 as shown in Figure 3.
- Synonymy: is a semantic relation that occurs between terms that share the same meaning. Synonyms are grouped together to form synsets. For example *standard* and *criterion* are synonyms in Figure 3.

Using this taxonomy, several semantic relatedness measures can be used to compute the closeness between any two terms in the hierarchy. A semantic relatedness measure (also called distance or similarity measure in literature) is a criterion to scale the relatedness of two concepts. Several factors were involved in the computations such as the depth of any two concepts from the root node, the common subsumer (node) between the two concepts in the hierarchy and the shortest path between the concepts.

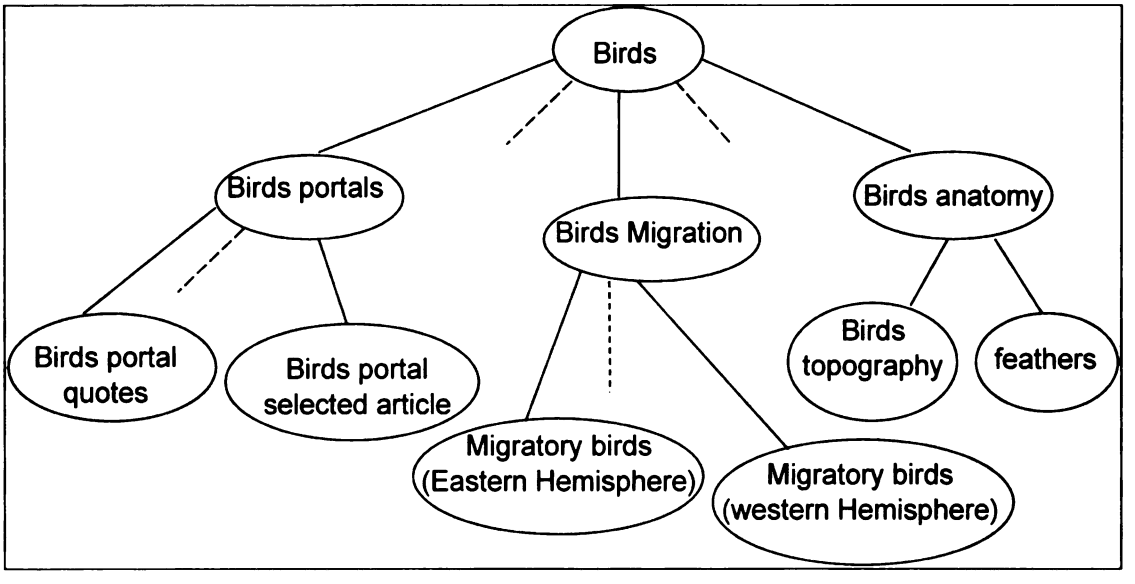
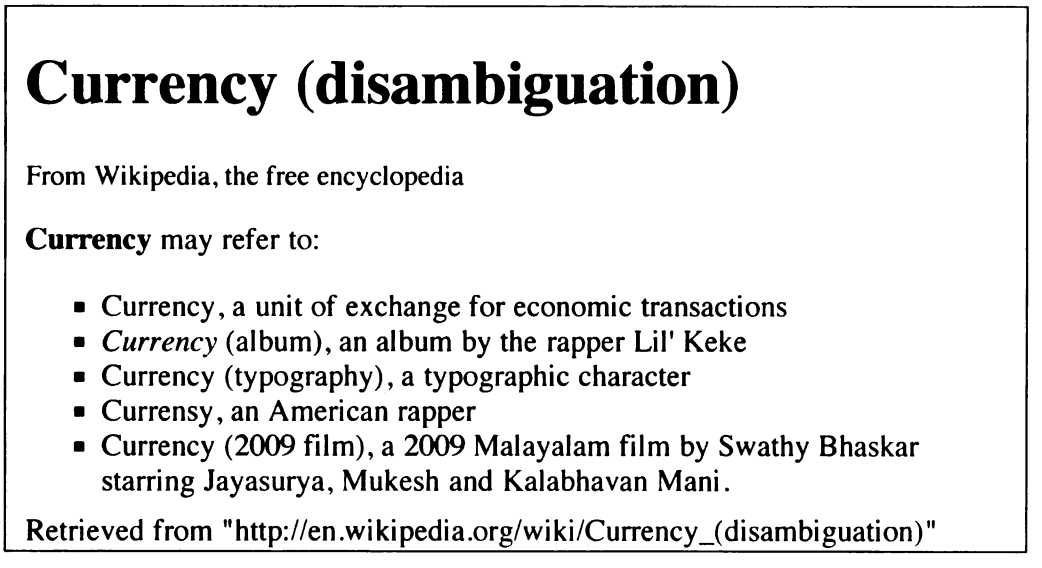


Figure 4: Category-subcatagory structure in Wikipedia

2.3.1.2 Wikipedia

Wikipedia is a web-based collaborative and multilingual encyclopedia that contains approximately 13 million articles, of which almost 3 million are English articles [43]. It grows dynamically and quickly as a result of the collaboration of thousands of contributors. It can be used as a thesaurus because the articles are organized in a taxonomy of categories where higher categories in the hierarchy subsume relevant categories under a more general unit of meaning.

Figure 4 shows an example of the categories structure in Wikipedia. Each article in Wikipedia is mapped to a topic (concept) in the thesaurus. Similar concepts are associated together via redirected links to the *preferred* concept.



Currency (disambiguation)

From Wikipedia, the free encyclopedia

Currency may refer to:

- Currency, a unit of exchange for economic transactions
- *Currency* (album), an album by the rapper Lil' Keke
- Currency (typography), a typographic character
- Currensy, an American rapper
- Currency (2009 film), a 2009 Malayalam film by Swathy Bhaskar starring Jayasurya, Mukesh and Kalabhavan Mani.

Retrieved from "http://en.wikipedia.org/wiki/Currency_(disambiguation)"

Figure 5: The disambiguation page of "currency"

For example there is a redirect link from the concept "puma" to "cougar" in the taxonomy. This mapping is beneficial to explain the synonymy relationship between concepts. Moreover, Wikipedia contains disambiguation pages for the concepts to

account for all the possible senses or meanings for polysemous concepts. Figure 5 shows the disambiguation page of the noun "currency".

2.3.1.3 MeSH

MeSH [41] is the Medical Subject Headings ontology that consists of controlled vocabulary mainly used to index biomedical abstracts from MEDLINE (Medical Literature Analysis and Retrieval System Online) using PubMed search engine. PubMed search space contains 18 million medical articles and journals dating back to 1948. These articles are indexed using entry terms and descriptors from the MeSH taxonomy. Entry terms, that are synonyms, are grouped together to form descriptors.

Figure 6: Part of MeSH tree

For example, "Octopodiformes" as a descriptor has the following entry terms {"Octopoda", "Octopus", "Octopuses"}. Descriptors correspond to categories in

WordNet and Wikipedia. They are organized in a MeSH tree that represents the taxonomy where each descriptor or category is related to its subcategories. Figure 6 shows an example from the MeSH tree.

2.3.2 Clustering using Ontology

In this section we review the related work on incorporating the ontology information into document clustering. We explain how the semantic knowledge can be extracted from WordNet, Wikipedia and MeSH ontologies and how it can affect document clustering.

2.3.2.1 WordNet

Most of the existing research on document clustering using WordNet is based on the well-known Vector Space Model (VSM) in which a document is represented by a vector of terms. Current approaches fall into three categories; terms-only VSM algorithms [56], concepts-only VSM [79][104][86] and hybrid VSM [35][92].

In the hybrid approach, the initial document vector is augmented with concepts from WordNet as added features. For example, Hotho et al. [35] was one of the first people to incorporate the WordNet ontology into document clustering. They filtered the non-noun terms from the initial text corpus and mapped them to their corresponding senses from WordNet. Hotho et al. [35] also investigated different word sense disambiguation techniques for the nouns. In one case, all the senses and the hypernyms up to level τ in the hierarchy were used to disambiguate a noun, while in another case, only the first sense was used to disambiguate a noun. Experimenting with different values of τ, they discovered that including hypernyms up to level five in the hierarchy helps in improving the purity of the clusters when bisecting k-means was applied to the Reuters dataset. However this technique would considerably increase the number of features (the

final number of features was nearly doubled), which leads to the curse of dimensionality problem. Sedding et al. [92] extended Hotho's work by exploring the benefit of disambiguating the terms using their part of speech tags (PoS) (i.e nouns, verbs, adjectives etc.) along with concepts from WordNet. They compared the clustering results using two types of features. The first feature set contains the original terms annotated with their PoS. The second feature set corresponds to the original terms annotated with their PoS combined with their corresponding synsets and hypernyms. Unlike the work by Hotho et al., their conclusion was that incorporating synonyms and hypernyms with PoS, did not seem to improve clustering results. They suspected the poor results were due to noise introduced by incorrect senses retrieved from WordNet.

Jing et al. [56] did not augment the Vector Space Model (VSM) with concepts from WordNet instead, they modified the terms weights in the document vector using semantic relations between the terms in WordNet. Then they clustered the documents using the new weights. In their method, two terms are related if any of them appears in the synsets of the other (synonyms and hyponyms senses were considered only). Equation (2) shows how the weight x_{jk} of the term t_k in each document X_j is modified:

$$\widehat{x}_{jk} = x_{jk} + \sum_{l=1}^{m} \delta_{kl} x_{jl} \qquad (2)$$

Where δ_{kl} is greater than 0 if t_k and t_l are related and 0 otherwise. K-means was applied on different samples from the 20Newsgroups dataset where a relative improvement of approximately 15% in entropy was obtained.

The concepts-only approaches are divided into two types; LSI-concepts algorithms [104] and frequency-concepts algorithms [86]. In LSI-concepts algorithms the co-occurrence information is included. Latent Semantic Indexing (LSI) assumes that there is an underlying semantic model that is estimated using the document-term frequency matrix. Termier et al. [104] developed an approach that transforms a document vector of terms into a corresponding vector of concepts, taking into account the co-occurrence of terms in the text corpus. They first applied LSI to decompose the document-term matrix and obtain its' components (see Equation (1) for more details). Using the left singular matrix, they computed a (*term, term*) similarity matrix from which they formed a (*concept, concept*) similarity matrix by mapping the terms into their corresponding concepts in WordNet. They then clustered the concepts into groups called sense units, which were finally used as document features in clustering. Their algorithm was applied to the Reuters dataset but the results suggested that combining LSI with WordNet was not that useful. They mentioned that the weakness of their approach was due to the imbalanced weighting between the statistical (LSI) and semantic (WordNet) feature space. Another way of using concepts as components of the feature vector was introduced by Recupero [86]. He limited his feature vector size to 41 categories from WordNet. A category is a high level concept in the WordNet taxonomy such as artifact, attribute and act, Table 3 shows a list of the nouns categories in WordNet. Each category is assigned a weight that corresponds to the number of terms that belong to this category in a given document. For example, the terms "dog" and "sheep" belong to the same category "animal" and therefore they both increment the weight of the same vector entry. This approach is known as WordNet Lexical Categories (WLC). Bisecting k-means was

applied to the Reuters dataset using the WLC features. The results were better than using the original terms as features. Recupero extended this approach by composing a new set of features called element sets. He used the terms from the top categories to build the element sets. A term is placed in an element set only if it has a semantic hypernym or hyponym relation with all other terms in the element set, otherwise the term will be moved to a new element set.

Tops	Act	Animal	Artifact
Attribute	Body	Cognition	Communication
Event	Feeling	Food	Group
Location	Motive	Object	Person
Phenomenon	Plant	Possession	Process
Quantity	Relation	Shape	State
Substance	Time		

Table 3: Nouns lexical categories in WordNet

Table 3 summarizes the results of incorporating ontology in document clustering as reported in the reviewed papers. In summary, previous research suggested that incorporating semantic knowledge from WordNet might help or hurt clustering. Wang et al. [112] also performed a comparative study to examine the effect of using concepts in document clustering. The Wilcoxon signed-rank test was used to investigate the benefit of augmenting the feature vector with senses from WordNet. They tested using different clustering approaches such as k-means, bisecting k-means and hierarchical clustering.

Reference	Data	Ontology	Measure	With Ont	Without Ont.
Fodeh et al.	Reuters	WordNet	Entropy	1.19	.97
	20Newsgroups		Entropy	.86	.88
Hotho et al.	Reuters	WordNet	Purity Inverse Purity	.61 .51	.57 .47
Wang et al.	Journals in AI	WordNet	Entropy F-measure	.56 .77	1.45 .45
	Journals/Ecology		Entropy F-measure	.52 .79	1.41 .45
	Economic		Entropy F-measure	.79 .66	1.55 .41
	Historical Abs		Entropy F-measure	1.09 .51	1.54 .42
Moravec et al.	Los Angelos	WordNet	Precision Recall	.73 1	.79 .74
Termier et al.	Reuters	WordNet	Accuracy	.81	.96
Sedding et al.	Reuters	WordNet	Entropy Purity	.74 .47	.59 .58
Banerjee et al.	Google News	Wikipedia	Accuracy	.85	.63
Yoo et al.	Medline	MeSH	Entropy Purity F-measure Mutual inf.	.12 .93 .84 .1	.13 .92 .80 .14
Hu et al.	Reuters	Wikipedia	Purity Inverse Purity	.65 .59	.60 .54
	OHSUMED		Purity Inverse Purity	.45 .38	.41 .34

Table 4: Results of incorporating ontologies in document clustering. Ont. is short of Ontology

Their experiments suggest that hierarchical clustering using concepts give better results. But since the method is computationally expensive, it may not be suitable for large scale datasets. In general, their conclusion was that including concepts might help in improving the clustering results, but the statistical improvement may not be statistically significant.

2.3.2.2 Wikipedia

The enriched thesaurus of Wikipedia attracted researchers to use its contents in text mining techniques such as classification [28][109][110], information retrieval [65][71], and document clustering [1][47]. Nevertheless, this integration can be very challenging because improper incorporation of concepts from Wikipedia may mislead the text mining process. One approach to extract concepts from Wikipedia is developed by Gabrilovich et al. [28]. Their approach is called Explicit Semantic Analysis (ESA) which represents the documents in a new high dimensional semantic space where the dimensions are the weighted concepts in Wikipedia. They first construct an attribute vector (v_i) for each Wikipedia concept that consists of the terms that appear in its corresponding article. The words' weights are normalized using the TFIDF scheme [90] to quantify the strength of associations between terms and concepts. An inverted index which maps each term to a list of concepts is then created to speed up the concept search when building the semantic representation of the text. An entry in the inverted index k_j^{wi} reflects the strength of the relation between the term w_i and the concept c_j. For a given text fragment T={w_i} ,using the inverted index, they retrieve all the concepts associated with the terms to compose the concepts semantic vector of the text fragment. The length of the concepts vector is equal to the total number of concepts in Wikipedia. Then each concept c_j is defined as $c_j = \sum_{wi} v_i \bullet k_j^{wi}$. These weights reflect the relevance of Wikipedia concepts to the text. Hue et al [49] extended Gabrilovich's approach and utilized the

discovered concepts into document clustering. Nonetheless, they used only the top 5 concepts per word from the inverted index in an effort to reduce the high dimensionality observed in Gabrilovich's approach. Further, they extracted the direct categories of the concepts and linearly combined them with the concepts and the original words to compute similarities between pairs of documents. Based on their conclusion, the categories add value to document clustering. Banerjee et al. [1] also used the contents of the Wikipedia articles to enrich the documents representation with concept to improve document clustering. To achieve this, they used the Lucene index [39]. Their study focused on short text documents where they created two queries for each document. One query corresponds to the title of a given document and the other contains the text. They retrieved the top 10 matching concepts from Wikipedia for each query from the Lucene index [39]. They augmented the document term vector with these concepts and applied the following six different clustering algorithms provided by CLUTO [38]. These algorithms include "direct" k-means, ("rb", "rbr") which are based on bisecting k-means, ("agglo", "bagglo") which are based on agglomerative clustering and the "graph" algorithm in which the objects are modeled in a graph. The experimental results showed that enriching the documents' representation with Wikipedia concepts achieved an improvement on Google News dataset. Despite the reported improvement (9% over using the original nouns and standard k-means), this method only used the concepts without considering the semantic relations in from the network.

Another way of using Wikipedia in text mining is to compute semantic relatedness between the concepts based on the contents as well as the network. Huang et al [50] used the contents of the articles as well as the hyperlink network between the

articles to extract concepts for Wikipedia. For a given document, they mapped all possible N-grams to the list of anchor texts in Wikipedia. (an anchor text is the text linked to a hyperlink). They use methods from [75] by Mihalcea et al to filter the anchor text and retain the ones that they use as concepts. The concepts are further pruned using a relatedness measure proposed by Milne [73]. In this measure, the relatedness between two concepts is based on the incoming hyperlinks to those concepts. Unlike Huang et al, Wang et al [110] built a thesaurus for this purpose. The thesaurus contains concepts from Wikipedia and explicitly derived relationships between the concepts such as: synonymy, polysemy, hyponymy and associative relationships. They used the article titles from Wikipedia as their concepts in the thesaurus. Furthermore they defined a comprehensive semantic relatedness measure to reflect the similarity between a pair of concepts. This measure is a linear combination of three different semantic relatedness measures:

- a content-based measure, in which the articles in Wikipedia are represented in the term vector space using TFIDF, and the semantic distance between any two concepts, is measured by computing the cosine similarity between their corresponding article vectors.
- an out-link category based measure, in which a vector of out-link categories is formed for each article in Wikipedia. Out-link categories of an article are "the categories that out-linked articles of the original article belong to" [110]. The semantic relatedness between any two concepts is the dot product divided by the cross product of their corresponding category vectors.
- a distance based measure, in which the semantic distance between a pair of concepts is measured by counting the number of nodes along the shortest path in

the taxonomy normalized by the maximum depth of the concepts in the taxonomy.

Unlike Banerjee et al., Hu et al. [47] computed the semantic relatedness between the concepts in Wikipedia. They used the same thesaurus introduced in [110] but with a slight difference on the semantic relatedness measure. They also compared the effect of their method on enriching documents with concepts from Wikipedia with Gabrilovich [28] method and Hotho's [35] method who used WordNet as a resource of semantic information. Evaluating their method on Reuters and OHSUMED datasets, they reported an improvement in the purity of the clusters over the other two methods. A relative improvement around 10% was reported over the baseline.

Comparing WordNet and Wikipedia, several researchers [28][65][110] claimed that WordNet is a limited taxonomy that does not provide sufficient information to solve the disambiguation problem in the text. That is because the 'gloss' (a small text description of a term in the taxonomy in WordNet) may not be sufficient to explain all the contextual uses of a concept as opposed to Wikipedia in which full articles are linked to the concepts to explain their contextual uses.

2.3.2.3 MeSH

Mesh has been used to aid clustering medical abstracts in Medline. Current approaches fall into two categories; graph-based algorithms [115] and VSM algorithms [95]. Yoo et al. [115] introduced a semantic-based bipartite graph representation for the documents. A bipartite graph is constructed where the two sets on the graph are the documents and the selected MeSH concepts. Terms in Pubmed abstracts are first mapped to the MeSH terms which are, in turn, mapped to the concepts (descriptors) of MeSH. The set of identified

concepts are used to construct the semantic feature units called co-occurrence concepts where two concepts are combined into a co-occurrence concept if they occur in the same document. A similarity matrix is then generated that encodes the similarities between pairs of co-occurrence concepts. The similarity between two co-occurrence concepts is computed by linearly combining their document coverage and the semantic similarity between their concepts using the MeSH tree. To get the clusters of the co-occurrence concepts, average-link clustering is applied to the co-occurrence concepts. The resulting clusters of the co-occurrence concepts were used to assign the documents to their clusters such that each document was mapped to the co-occurrence cluster that contained the bigger number of co-occurrence concepts existing in that document. According to the authors, one of the benefits of using the taxonomy was alleviating the synonymy problem when bridging between the terms in Pubmed abstracts and their corresponding concepts (descriptors) in MeSH via the entry terms.

Using a VSM representation of the medical abstracts, Shin et al. [95] combined the original text of the abstracts with the MeSH entry terms that annotate them. Those terms in the abstracts that are related to the MeSH terms, were given more importance by adjusting their weights in the feature vector. Their experimental results showed that augmenting the medical abstracts with MeSH terms improves the clusters quality. An improvement of (36 to 47)% was observed when MeSH terms were included. Unlike Shin et al., Zhu et al. [122] did not combine the MeSH concepts and the medical abstracts contents instead, they first constructed a similarity matrix from each source and linearly combined them. An entry X_{ij} in the content similarity matrix corresponds to the content

similarity between a pair of documents (d_i, d_j) which is computed using the cosine similarity of their corresponding term vectors. On the other hand, an entry in the semantic similarity matrix Y_{ij} encodes the semantic similarity of (d_i, d_j) which is computed using the MeSH concepts that are associated with each document. More specifically, Y_{ij} is set to be the sum of the semantic relatedness between all pairs of Mesh concepts of both documents. Both matrices are then linearly combined to form the integrated matrix. A spectral clustering algorithm is applied on the integrated similarity matrix to produce the document clusters. Their experimental results showed a significant improvement in the quality of the obtained clusters.

2.4 Summary

We have presented a survey about incorporating ontology into document clustering. The conclusion from previous research is not conclusive. While some researchers concluded that incorporating ontology in document clustering does not help, others showed that it can help. Therefore more research in needed in the field.

Chapter 3

3 Data and Preprocessing

In this chapter, we give a detailed description of the benchmark datasets we have used for evaluating our proposed clustering algorithms. We also describe common preprocessing steps performed on documents as well as concept mapping approaches developed to incorporate ontology to convert the original words feature vector into its corresponding vector of concepts.

3.1 Data

We have used several benchmark datasets to evaluate clustering algorithms, namely 20Newsgroups [40] and Reuters-21578 [36] datasets. We generated different samples from each dataset. This section presents the details of these samples.

3.1.1 Reuters-21578

Although Reuters-21578 [36] has been widely used for evaluating document clustering algorithms, it has several known limitations. While many of the documents are not labeled, others are assigned to multiple classes. Furthermore, it has a skewed distribution of the documents. The following samples have been used in the literature to deal with these problems and to evaluate clustering algorithms.

Min15-Max100: This sample has been used by Hotho et al. [35] and Sedding et al [92]. The topics included in this dataset have at least 15 documents and at most 100 documents. Because of this restriction, the number of topics retained was 46. Documents with multiple labels have been included by assigning them to the first label.

Min20-Max200: this is similar to the min20-max200 sample used by Huang et al [50] and Hu et al [47]. It covers 30 topics where each topic contains documents in the range of 20 to 200. Only single labeled documents were included in this dataset.

In addition, we sampled the following datasets to evaluate our algorithms:

Rueters1: this dataset contains the top 20 largest classes. The unlabeled documents are discarded but the documents with multiple labels are assigned to their first class as long as it belongs to the top 20 largest classes (see the Appendix for the list of topics). It contains a total of 9456 documents, 32166 words and 8276 nouns.

Rueters2: We randomly sampled a maximum of 200 documents from each class. The final size of this dataset is 2655 documents.

3.1.2 20Newsgroups

20Newsgroups dataset [40] contains articles from 20 different newsgroups, each corresponding to a different class. The original data contains 18846 documents. We created four samples from the original data, which are called *20NG-100*, *Multiple-6000*, *Binary-Multiple-500* and *Binary-400*.

20NG-100: This dataset was created following the steps described in Hu et al [47]. All 20 classes of *20Newsgroup* are included. From each class, we sampled 100 messages. Stemming is not applied to the documents in order to maintain consistency with the sample dataset used in [47].

Multiple-6000: Unlike Reuters-21578, 20Newsgroups does not suffer from the problem of documents having no labels or multiple labels. However, since some of the classes are highly correlated, the 20 classes are aggregated into 6 distinct classes, as shown in Table 5. For example, the subclasses "rec.autos", "rec.motorcycles", "rec.sport.baseball", and

"rec.sport.hockey" are aggregated into the class "recreation". In this dataset, we randomly sampled 6000 articles from the training set where each class contains 1000 article. Table 5 shows a list of these classes.

Class Id	Class	Description
1	Computer	comp.graphics
		comp.os.ms-windows.misc
		comp.sys.ibm.pc.hardware
		comp.sys.mac.hardware
		comp.windows.x
2	Recreation	rec.autos
		rec.motorcycles
		rec.sport.baseball
		rec.sport.hockey
3	Science	sci.crypt
		sci.electronics
		sci.med
		sci.space
4	Miscellaneous	misc.forsale
5	Politics	talk.politics.misc
		talk.politics.guns
		talk.politics.mideast
6	Religion	talk.religion.misc
		alt.atheism
		soc.religion.christian

Table 5: 20Newsgroups Classes

Binary-Multiple-500 dataset:

This dataset was originally created by Slonim and Tishby [106]. A similar version of the data was used by Dhillon et al. [17]. There were altogether 9 datasets generated by sampling documents from 2, 5 and 10 sets of classes. Each dataset has 500 documents that are evenly distributed across the classes. The classes used for the three binary datasets are talk.politics.mideast and talk.politics.misc classes. The multi5 datasets are created by sampling from the sci.space, comp.graphics, rec.motocycle, rec.sports.baseball and talk.politics.mideast classes while the multi10 datasets are generated from the following classes: sci.med, sci.space, sci.crypt, rec.autos, misc.forsale, alt.athiesm, sci.electronics, rec.sport.hockey, talk.politics.gun and comp.sys.mac.hardware. Table 6 shows the number of terms and the number of senses after disambiguation using the method described in Section 3.3.

Dataset	#Words	#Nouns	#Senses	#Classes	#Documents
Binary1	15496	6023	6148	2	500
Binary2	15526	6052	6224	2	500
Binary3	15660	6131	6274	2	500
multi51	13955	5388	5700	5	500
multi52	14902	5677	5930	5	500
multi53	14126	5601	5849	5	500
Multi101	13946	5578	5875	10	500
Multi102	12884	4379	5563	10	500
Mulit103	13615	5177	5834	10	500

Table 6: Number of different features in Binary-Multiple-500 dataset

Binary-400 dataset:

This is a collection of 16 binary datasets (bin1, bin2, ..., bin14) that were generated from the 10 classes used in the multi10 dataset. Each binary dataset contains 400 documents, out of which 200 are randomly sampled from each class. The average number of terms is approximately 4000 for each dataset.

3.2 Preprocessing

An ordinary document may contain some non-descriptive and redundant words that may degrade the document clustering performance. Therefore, the documents must be preprocessed to remove these uninformative words. Typical preprocessing includes the following steps:

- Stopwords removal: drops non-descriptive words from the text. Examples are: pronouns, determinates and numbers. We used one the stopwords lists that is publically available in [42]. Stopwords removal is not critical because the noun lexical file of WordNet does not include most of the stopwords (with a few exceptions such as months, days of the weak, numbers and years). Therefore, we augmented the list and included such words. The list of stopwords we have used has 648 entries. (see Appendix).
- Stemming: reduces a given word to its roots. Porter stemmer [84] and Morphy Stemmer [80] are two common stemmers that have been intensively used in the literature. In our work, the nouns that we retain as features are stemmed using the Morphy Stemmer. The flowchart in Figure 7 illustrates the sequence of the preprocessing steps applied to the different datasets.
- Normalization: scales the frequencies of the words within/across the documents. Term Frequency Inverse Document Frequency TFIDF shown in Equation (1) is a

common method proposed by Salton and Buckley [90] to normalize the documents. It weighs the strength of the association between that word and the document it appears in. TFIDF can improve the clusters quality by 14% as mentioned in [90].

$$tfidf_{d_j}^{w_i} = tf_{d_j}^{w_i} \log \frac{N}{Df^{w_i}} \quad (1)$$

Where w_i^j : i^{th} word in the j^{th} document.

$tf_{d_j}^{w_i}$: word frequency of the i^{th} word in the j^{th} document.

Df^{w_i} : number of documents that contain the word w_i

N : number of documents in a dataset.

In our thesis, we apply TFIDF to all datasets in both the noun-based representation and concept-based representation.

As shown in the flowchart in Figure 7 the next step after ordinary preprocessing is concept mapping. Essentially, this operation is needed to map the original words in the documents to their corresponding concepts in the ontology being used. As we extract the concepts from two different ontologies WordNet and Wikipedia, we have two different methods for concept mapping, which we illustrate in the following sections.

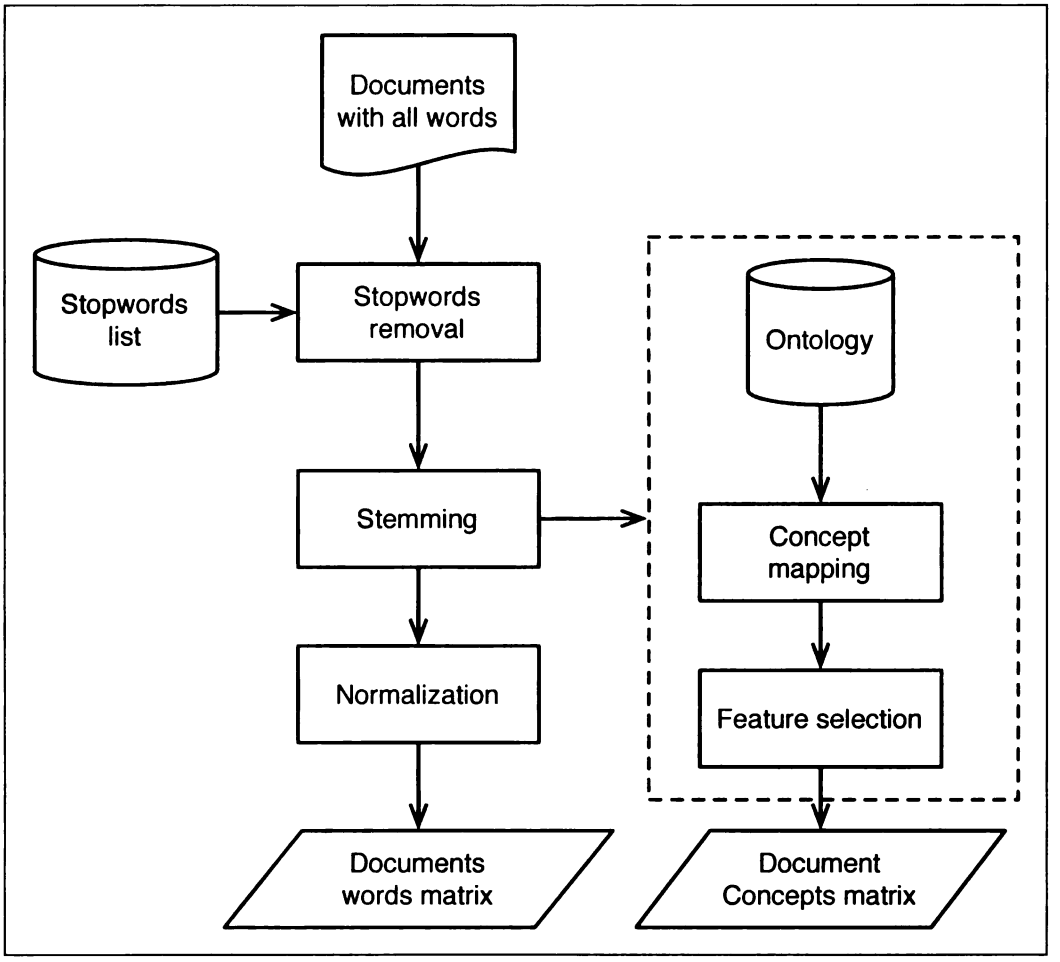


Figure 7: Flowchart of the reprocessing steps applied to the original documents

3.3 WordNet-Based Concept Mapping

A key step to incorporate semantic knowledge from WordNet by identifying the most appropriate sense associated with each noun in a given document is concept mapping which is also known as Word Sense Disambiguation. Our approach is focused on disambiguating nouns based on the "context" of other nouns within a document. For example, consider the term "cat", which has eight meanings as a noun in WordNet. If it is used in a document that contains other terms such as "kitten", "Persian", and "pet", we expect its sense refers to the "feline mammal" sense of cat, and not one of the other seven (such as a farm machine or a particular type of X-ray). However, if the term "cat" appears

in a document that contains other nouns such as "construction", "builder", and "home", its sense most likely refers to a "Caterpillar", which is a large tracked vehicle used for moving earth in construction and farm work.

More formally, our approach can be described as follows. Let S_i = $\{s_{i1}, s_{i2}, \ldots s_{ik}\}$ denote the set of all senses associated with the noun t_i according to the WordNet ontology. Given a document d, we determine the most appropriate sense $\widehat{s}_i$ of a noun t_i by computing the sum of its similarity to other noun senses in d, i.e.

$$\widehat{s}_i = \arg\max_{s_{il} \in S_i} \sum_{t_j \in d} \left[\max_{s_{jm} \in S_j} \sigma(s_{il}, s_{jm}) \right] \qquad (2)$$

where $\sigma(s_p, s_q)$ is the WordNet similarity between two senses, s_p and s_q. Furthermore, since the senses of a given noun in the WordNet hierarchy are arranged in descending order according to their common usage, we restrict our consideration to the first senses of the noun. As given in the WordNet hierarchy, we selected the top 3 senses since we noticed through our experiments that using all senses does not make a significant change in the results. Therefore, restricting the number of senses to be examined to 3 reduced the computational complexity of our approach. Despite the fact that term sense disambiguation using ontologies can be quite expensive, it is only a one time cost. It is done once per document and the extracted senses are stored to be used at any time.

Furthermore, this disambiguation process can be done in parallel since the task is done independently for each document.

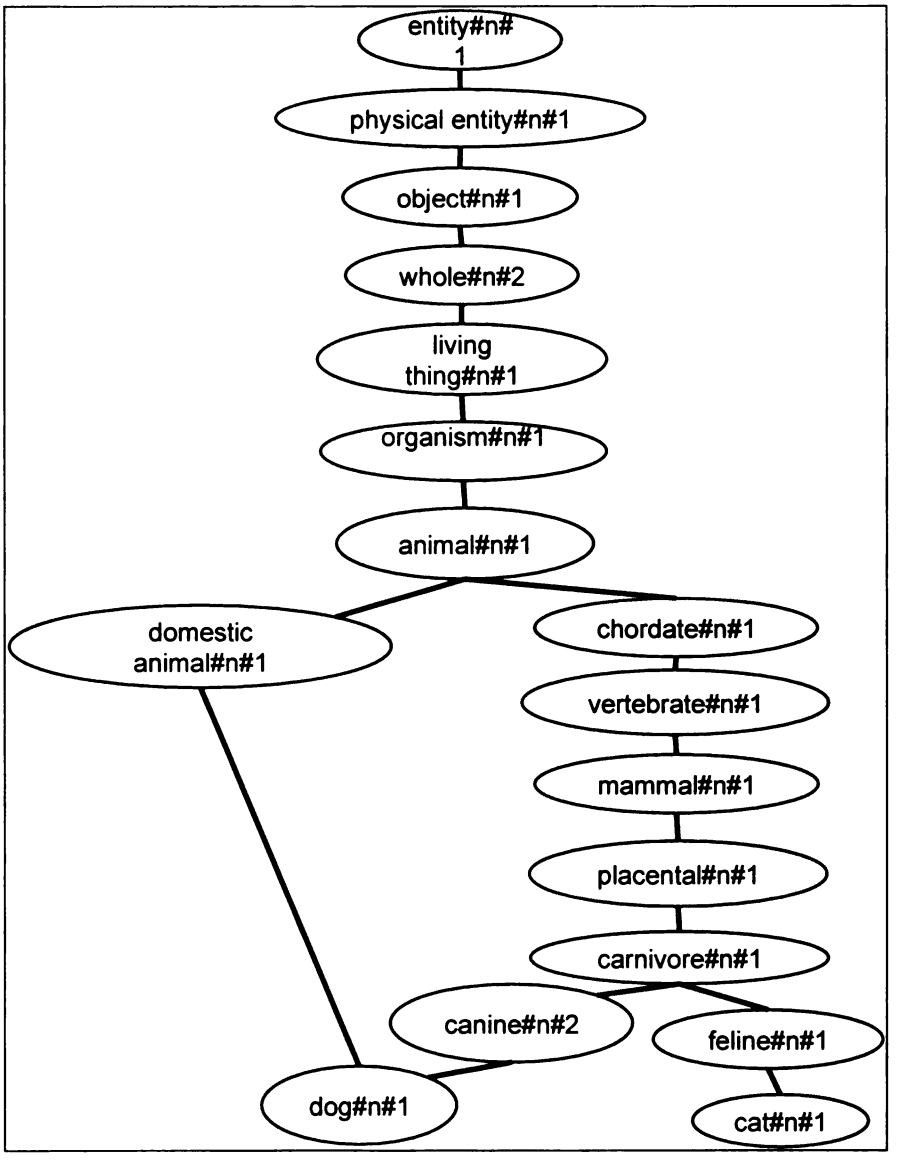


Figure 8: semantic network between the first senses of dog and cat

Several similarity measures $\sigma(s_p, s_q)$ can be used in Equation (2) to compute the similarity between two senses such as *Wu-Palmer* [114], *lin* [66] and *jcn* [54]. However, we used Wu-Palmer similarity measure [114] in our experiments. Wu-Palmer is a path-

based measure which considers the shortest path between the concepts and their depth in the hierarchy. It computes the similarity between two senses by finding the least common subsumer (LCS) node that connects their senses.

$$\sigma_{\text{Wu-Palmer}}(s_p, s_q) = \frac{2d}{L_p + L_q + 2d} \tag{3}$$

where L_p is the path length between S_p and the LCS, L_q is the path length between S_q and the LCS and d is the depth of the LCS from the root. The depth of the LCS is computed in [44] as:

$$D(LCS) = path_length(\text{LCS}, root) + 2 \tag{4}$$

Figure 8 shows part of the WordNet hierarchy that associates the first sense of the nouns "dog" and "cat". There are two common subsumers between both senses which are "animal" and "carnivore". Wu-Palmer's strategy favors the Lowest Common Subsumer in the hierarchy between the two senses. In this case, "carnivore" is the LCS because the depth of "carnivore" is 13 whereas the depth of "animal" is 8 in the hierarchy.

$$\sigma_{\text{Wu-Palmer}}(dog\#1, cat\#1) = \frac{2x13}{2+2+2x13} = .866$$

The Wu-palmer similarity score between "dog" and "cat" using "carnivore" as the common subsumer is .866 which is higher than the similarity score .64 that we obtain when using "animal" as subsumer.

Using Wu-Palmer in equation (3) produced the semantic features (concepts) that correspond to the original nouns. The semantic representation of the documents is then

established by creating a binary vector for each document where the existence of a concept in a document is denoted by 1 and 0 otherwise.

3.4 Wikipedia-Based Concept Mapping

Wikipedia has been recently used to generate a concept-based representation for a collection of documents [28][47][54]. A critical step is to preprocess the Wikipedia content, namely, its articles and categories, prior to extracting the concepts. In this study, we used the Wikipedia data dump from Oct. 28, 2009 for our experiments. The Wikipedia data is publicly available in XML format at http://download.wikipedia.org. After preprocessing the XML file, we extracted 3,076,819 articles (concepts) and 519,039 categories in the category structure. This section presents the preprocessing steps applied to Wikipedia articles and the concept mapping approach used to transform a collection of documents into their concept-based representation.

3.4.1 Preprocessing Wikipedia Articles

The concepts in Wikipedia are titles of articles (i.e., Web pages) contributed by Wikipedia editors. The textual content of each article typically provides more detailed description about the concept, including its meaning and context of usage. Because of their 1-to-1 correspondence, we use the terms "concept" and "article" interchangeably in the remainder of this thesis. The following steps are applied to preprocess the Wikipedia data:

1. Each article is normalized to have unit length to address the problem of varying length articles. Some of the lengthy Wikipedia articles contain a large number of words that do not directly convey the meaning of the concept represented by the article. For example, the Wikipedia article on "algorithm" contains words such as

grain, yield, and insect, which are unrelated to the concept of "algorithm". These words however are relevant to the concept description of another Wikipedia article on "agriculture". Normalizing the articles to unit length would alleviate the problem by adjusting the weights (frequencies) of the words so that they reflect the importance of each word in a given article. As an example, the words grain, yield, and insect appeared once in the "algorithm" article (which has 2393 unique words with a sum of squared frequency equals to 10359). Upon normalization, the weight of each such word reduces to 0.0098. In contrast, their weights in the "agriculture" concept remain high even after normalization: grain (0.1158), yield (0.1336), and insect (0.0356).

2. To facilitate rapid access of the relevant Wikipedia concepts that contain a given word, we create an inverted index of the words. This index encodes the weight of a particular word w_i in each concept c_k in Wikipedia. The weight assigned to each word in a given concept is the normalized TFIDF score, which is denoted as $tfidf_{c_k}^{w_i}$. Analogously, a normalized $tfidf_{d_j}^{w_i}$ score is also computed for each word w_i in a given document d_j to be clustered.

3. To construct a concept hierarchy, the Wikipedia category graph is preprocessed to remove cycles. This is achieved in the following way. First, we apply breadth first search to compute the depth of each category, starting from the root concept (articles). Starting from each concept, we then traverse the graph in a bottom-up fashion to extract the higher-level categories that are linked to each concept.

3.4.2 Document Concept-based representation

In this section, we describe our method of generating a concept-based representation of a collection of text documents. Our method is an extension of Gabrilovich et al approach. In this method, a relatedness measure between a concept in Wikipedia and a document is established based on the textual contents in both text fragments. More specifically, the relatedness of a Wikipedia concept/article to a text document is quantified by the dot product between their corresponding word vectors in the feature space. The significance of the relatedness is proportional to the value of the dot product. The higher the value of the dot product, the stronger is the semantic relatedness. Formally, the relatedness of a concept c_k to a document d_j is the value of the dot product between their respective vectors as shown in equation (5).

$$c_k^{d_j} = \sum_{w_i \in d_j} tfidf_{d_j}^{w_i} \bullet tfidf_{c_k}^{w_i} \qquad (5)$$

The relatedness score of the concepts in all the documents can be computed via a matrix multiplication operation, as illustrated in the diagram below:

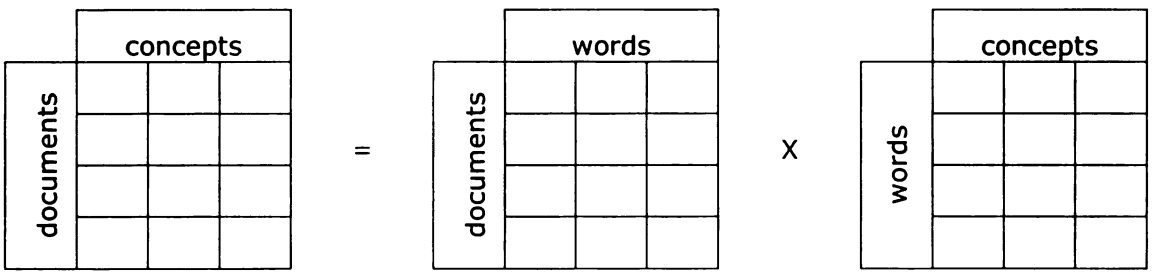


Figure 9: Relatedness score computations via matrix multiplication

One of the limitations of this approach is the high dimensionality of the document-concept matrix, which potentially includes irrelevant concepts. As reported in [28], the final set of concepts obtained contains 241,393 concepts used to represent the documents. A more effective strategy would be to prune the unnecessary concepts that are weakly related to the original document. Therefore, the documents should be ranked first based on their relatedness to a give document. Hu et al [49] proposed to rank the concepts based on the relatedness score such as computed by Equation (5) and dropping all but the top 200 concepts for each document. However, we argue that such ranking, which is solely based on the dot product of the TFIDF weights of the words in the concept and document vectors, is not sufficient. That is words in a concept with high TFIDF weight, yet not related to the topics in the concept, increases the relatedness score of the concept, hence amplifies its rank. To address this problem, we modified the relatedness score of a concept to a given document by the frequency of mapping that concept to the document. This frequency is equivalent to the number of common words between the document and the concept as the numerator in equation (6). It is normalized by the total frequencies of all other concepts mapped to the same document as shown in the denominator in equation (6):

$$w(c_k, d_j) = \frac{\sum_{w_i \in d_j} I(w_i \in c_k)}{\sum_{c_k \in C_{d_j}} \sum_{w_i \in d_j} I(w_i \in c_k)} \qquad (6)$$

Where C_{d_j} is the set of concepts mapped to the document d_j and I($w_i \in c_k$) is 1 if the w_i exists in concept c_k and 0 otherwise. Therefore the relatedness score of a concept to a document is modified as:

$$c_k^{d_j} = w(c_k, d_j) \sum_{w_i \in d_j} tfidf_{d_j}^{w_i} \bullet tfidf_{c_k}^{w_i} \qquad (7)$$

This updated relatedness score prefers concepts that are more semantically close to the original document. Previously, the relatedness score as computed by equation (5) was sensitive to the TFIDF weights of the words in c_k. If a word has a high TFIDF weight in c_k, it will cause the concept to be a strong candidate in the final feature set irrespective of its semantic relatedness. As a result, noisy concepts can be easily included in the final feature set. Nonetheless, scaling this score with the relative frequency of the concept causes the concepts that enclose more words to more likely appear as related. Thus the method will reduces the possibility of mapping totally irrelevant concepts to the document but does not completely eliminate it.

List of words in the document		
Article	Buy	Deal
Dealer	expense	Money
Net	Pay	People
Profit	Price	Car
Saturn	State	Time

Table 7: List of some of the words that appear in the document *d*

List of top concepts using Gabrilovich approach
timeline of the united states inventions
general motors
History of economic thought
Plug-in hybrid
Economy of the people's republic of china
Fairtax
ethanol fuel

Table 8: Top concepts mapped to the document *d* using Gabrilovich approach

To illustrate the effectiveness of our concept mapping approach compared to Gabrilovich method [28], consider the document *d* from 20Newsgroup dataset that belongs to "rec.auto" topic. This document is mainly about "cars". Table 7 shows a list of the words that appear in that document. Some of the top concepts extracted using Gabrilovich approach appears in Table 8 whereas Table 9 shows the top concepts generated using our method. High concepts in the list are more related to the document. Note that the concepts extracted by Gabrilovich approach contain only the concept "General Motors" that is directly related to the topic of the document. On the other hand, most of the concepts generated by our method are related to "cars" such as "Saturn Corporation", "Sega Saturn" and "Car Dealerships in North America".

List of top concepts using our approach	
saturn corporation	18
Saturn	16
Car dealerships in north America	27
sega Saturn	22
saturn aura	12
saturn (mythology)	13
the plain dealer (newspaper)	17

Table 9: Top concepts mapped to the document *d* using our proposed approach

In addition these concepts have more words in common with the document *d* as shown in column 2 of Table 9. This implies that the concepts extracted by our method are more significant and more related to the topic of the document.

3.5 Summary

In this chapter, we have given a detailed description of the datasets used to test the algorithms and the preprocessing steps that we performed. We have also explained the concept mapping approaches used to extract the concepts from WordNet and Wikipedia.

Chapter 4

4 Effect of Preprocessing on Document Clustering

In this chapter we present the evaluation metrics and the baseline clustering algorithm used in the thesis. We then demonstrate the effect of removing stopwords, noun identification via WordNet and stemming on document clustering. Finally we present a feature selection approach to mine semantic features with high information gain and its effect on document clustering.

4.1 Evaluation Metrics

We evaluate the proposed methods in the thesis using three different measures Purity, Entropy, and Normalized Mutual Information NMI.

Purity [119]: assumes one to one correspondence between the clusters and the classes. It focuses on the class that has the most frequency in each cluster. Purity is computed by calculating the weighted average of the maximum precision values of all clusters. High purity scores indicates good clusters.

$$Purity = \sum_i \frac{|C_i|}{n} \max(precision(C_i, L_j)) \qquad (1)$$

where the precision(C_i, L_j) of a cluster C_i for a class L_j is defined as:

$$precision(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|} \quad (2)$$

Entropy: measures the uniformity of a cluster. The entropy of all clusters is computed by calculating the weighted sum of the entropy of all clusters. Low value of Entropy indicates good clusters.

$$Entropy = -\sum_i \frac{|C_i|}{n} \sum_j p_{ij} \log p_{ij} \quad (3)$$

$$p_{ij} = \frac{1}{|C_i|} \sum_k I(d_k, L_j) \quad (4)$$

Where $I(d_k, L_j) = 1$ if the label of d_k equals L_j.

NMI [119]: measures the normalized mutual information between the cluster assignment and the ground truth of the documents. NMI is an increasingly popular measure of cluster quality.

$$NMI(X,Y) = \frac{I(X;Y)}{(\log k + \log c)/2} \quad (5)$$

where X, Y are random variables of the assignments of the clusters and the ground truth labels. K is the number of clusters and C is the number of the labels. The higher the value of NMI; the better are the clusters.

4.2 Baseline Clustering Algorithm

We use spherical kmeans as the baseline clustering algorithm to evaluate the proposed methods in this thesis. Spherical kmeans is a popular clustering method for clustering high dimensional text data in which cosine similarity measure is used to compute the angle between documents and centroid vectors. The time complexity of the algorithm is linear in the number of documents and clusters.

$$\cos ine(X,Y) = \frac{X \bullet Y}{\| X \| \ \| Y \|} \tag{6}$$

Despite its wide use, it suffers from the initial centroids problem. That is the quality of the partitions obtained is highly dependent on the initial centroids. To address this problem we run spherical kmeans 50 times and report the purity of the solution that better minimizes the objective function.

4.3 Effect of Preprocessing

As mentioned in Chapter 3, the preprocessing steps performed on the documents include stopwords removal, nouns identification by WordNet and stemming. Some of the advantages of using our method of preprocessing are:

- Simplicity. We reduce number of steps required for preprocessing. We skip part of speech tagging, instead using only stemming and noun identification via WordNet. The WordNet noun identification step is a lookup.
- Efficiency. Our approach effectively reduces the size of the feature space; instead of using all words, we only use the nouns as identified by WordNet as input for the clustering algorithm.

In this section we do an extensive experimental study to show the adequacy of using nouns in document clustering. We generate a baseline clustering solution using spherical kmeans and the nouns as features. Our goal is to show that the noun-based approach often yields a more rigorous baseline than using all words. We compare the baseline performances reported in some of the research papers mentioned in Chapter 2 against our simple baseline. In addition, we include the performance of the new methods proposed in the respective papers. We followed the steps of the authors as best as we could to create an equivalent baseline. Column 3 in Table 10 shows the purity obtained using the proposed clustering methods in the mentioned papers. Column 4 shows the reported purity using the baseline. Comparatively, column 5 lists the purity obtained using our proposed baseline using the nouns as identified by WordNet. We have also attempted to re-generate the baseline purity values reported in the papers. For the most part, the differences in their reported and our regenerated baseline are small, except for the dataset used by Huang et al [50] and Hu et al [47]. They reported a purity of 0.603 for their baseline compared to 0.73 produced by our words baseline. We presume that these differences come from either implementation or other detailed matters. For example, different implementations of Porter stemmer and different stopwords list could produce a different set of features in terms of size and contents. Furthermore, it is not clear that TFIDF normalization was applied to Reuters2 dataset for the reported baseline. When w applied the reported method without TFIDF, we obtained similar results to what was reported.

The performance of our proposed baseline was significantly better than the reported baselines as shown in Table 10. This is true across all methods. For example, for the 20NG-100 dataset, Hu et al [49] reported a purity of 0.411 using all words as

Reference	Data	Reported		Nouns Baseline
		Method	Baseline	
Hotho et al*	Min15-Max100	0.61	0.571	0.604
Sedding et al*		0.47	0.58	
Deerwester et al		0.596		
Hu et al.	Min20-Max200	0.65	0.603	0.791
Huang et al		0.678		
Deerwester et al		0.748		
Hu X. et al	20NG-100	0.442	0.411	0.52
Slonim and Tishby	Binary1**	0.7	0.62	0.92
	Binary2**	0.68	0.57	0.894
	Binary3**	0.75	0.65	0.92
	Multi51	0.59	0.38	0.924
	Multi52	0.58	0.36	0.912
	Multi53	0.53	0.34	0.943
	Multi101	0.35	0.24	0.742
	Multi102	0.35	0.27	0.724
	Multi103	0.35	0.28	0.734

Table 10: Comparison between the nouns baseline (column 5) against reported baseline (column 4) and reported clustering methods in the literature (column 3) in terms of purity. (*) This result was created using 60 clusters to make an equivalent comparison. (**) This result was created using 4 clusters

compared to 0.52 using our approach. The approach by Huang et al [50] and Hu et al [47] whose baseline used stemmed words had a purity of 0.603 compared to 0.79 using our baseline. More interestingly, the algorithms proposed by some of the authors gave poorer

results than those using our nouns baseline. Note that Hotho et al [35] partitioned the Min15-Max100 dataset using bisecting k-means into 60 clusters instead of 46, which is the number of topics in the documents. However, to be consistent with his experimental settings, we clustered the documents into 60 partitions using bisecting k-means. The result was a purity of 0.604 using our baseline compared to 0.571 using their baseline of stemmed words.

4.3.1 Why Nouns?

Results from the previous subsection suggest that the noun-based clustering approach

Class	%Nouns	Class	%Nouns
alum	0.75	livestock	0.90
bop	0.75	money-supply	0.80
cocoa	0.85	nat-gas	0.70
coffee	0.80	orange	0.75
copper	0.85	pet-chem	0.80
cotton	0.90	reserves	0.70
cpi	0.70	retail	0.80
gas	0.80	rubber	0.80
gnp	0.90	ship	0.90
gold	0.90	strategic-metal	0.85
grain	0.85	sugar	0.95
housing	0.80	tin	0.80
ipi	0.65	veg-oil	1.00
iron-steel	0.75	wpi	0.70
jobs	0.70	zinc	0.95

Table 11: Fraction of nouns among the top 20 words of each cluster centroid

often produces comparable and sometimes better clusters than using all words but with a significantly reduced number of features. This section provides further evidence to

indicate the importance of nouns in capturing the underlying structure of the data and its clusters.

4.3.1.1 Nouns and Centroids Analysis

In this experiment, we investigate the nouns role in forming the different topics in a

cocoa		natural gas		grain	
nouns	**words**	**nouns**	**words**	**nouns**	**words**
Cocoa	cocoa	gas	Ga	Grain	grain
Buffer	buffer	natural	Nature	Usda	usda
Stocks	icco	pipeline	pipelin	Soviet	crop
Delegate	Stock	foot	feet	crop	soviet
Rules	Deleg	energy	cubic	certificates	certif
Tonne	Tonn	contract	energi	agriculture	agriculture
Manager	Rule	texas	file	tonne	mln
Council	manag	company	contract	estimate	tonn
Consumers	purchas	customer	flow	official	estim
International	Bui	oil	compani	import	offici
Producer	council	corp	texa	department	ussr
Ghana	consum	access	coastal	program	report
Purchase	Intern	petroleum	transamerican	farm	harvest
Differential	ghana	bankruptcy	court	ussr	drought
Bra	produc	field	corp	harvest	farm
Buy	differenti	interest	mln	elevator	elev
Organization	Organ	filing	transco	china	lyng
Compromise	compromis	canadian	transport	bill	program
Season	kanon	subsidiary	custom	report	import
Meeting	Bra	spot	access	land	depart

Table 12: Comparison between the noun-based and word-based centroids

Table 12 (cont'd)

Gold		Copper		Jobs	
Nouns	**words**	**nouns**	**words**	**nouns**	**words**
Gold	gold	copper	copper	unemployment	unemploy
Mine	mine	cent	cent	Pct	Pct
Ounce	ounc	lb	Lb	workforce	workforc
Ton	coin	cathode	cathod	number	number
Coin	Or	price	effect	unemployed	unemploi
Foot	Warrant	alloy	immedi	Fell	Adjust
Ore	Explor	corp	pound	people	Fell
Tons	Venture	subsidiary	price	benefit	Season
miner	Compani	octane	mine	record	Mln
Mining	Dlr	magma	ct	Rate	Total
Fire	South	mining	corp	employment	People
Company	Product	aluminum	newmont	statistics	Benefit
Reserves	Issu	electrolytic	magma	labour	Jobless
Production	Corp	pound	subsidiari	Total	Statist
Venture	Bullion	lowering	alloi	figure	Rate
Deposit	Deposit	raising	electrolyt	office	Record
Exploration	Mln	company	compani	department	End
South	Price	pump	inspir	labor	Labour
Property	Miner	rod	lower	official	Employ
Underground	Property	mineral	resin	application	Compare

collection of documents. Our intuition was that the nouns are important in a cluster if They contribute significantly to the weights of its corresponding centroid when clustering using all the words. Our strategy was to show the importance of the nouns by examining the centroids of the clusters. Specifically, we select the top 20 words with highest weights in each cluster centroid and compute the fraction of nouns among the top 20 words.

The higher the fraction, the more important are the nouns in terms of defining the main theme of the cluster. We used Min20-Max20 to test our statement in this experiment. This dataset has 30 classes. Out of the 7170 words, about 64% of them are nouns. In Table 11, the first and third columns show the list of clusters, each labeled with the class that has the majority of documents in that cluster. The second and fourth columns show the corresponding fraction of nouns in the top 20 words of the cluster centroid. The results in Table 11 suggest that the nouns occupy a large portion of most of the centroids. For example, coverage above 85% is observed for cocoa, gold, sugar, and zinc topics. It is important to mention that having the nouns dominating the centroids does not necessarily assure a better solution. However, it does suggest that one may still be able to recover most of the clusters after removing the non-nouns from each document. Table 12 compares the top 20 features of 6 selected centroids obtained by applying spherical k-means on Min20-Min200 words and Min20-Min200 nouns. Note the high overlap between the top 20 features of the noun-based and word-based centroids.

4.3.1.2 Nouns and LSI Analysis

In previous experiments, we showed that the nouns are a major component in document clusters. In this experiment, we examine the importance of nouns in terms of capturing the overall structure of the data. We consider two approaches for evaluating this. The first

approach examines the percentage of nouns in the concepts produced by Latent Semantic Indexing LSI. LSI is a widely used method to find a low-rank approximation of the term-document matrix by decomposing it into matrices of left and right singular vectors. We examined the fraction of nouns that contribute to the first 10 components (i.e., left singular vectors). Specifically, we sort the words based on their magnitude in a given component and select the top words at 99^{th} {top 72 words}, 95^{th} {358 words} and 90^{th} {717 words} percentiles. Table 13 shows the results. Observe that the nouns contribute to at least 75% of the words in the 99^{th} percentile across all 10 components, concepts. This result again emphasizes the important role of nouns in identifying the topics in a dataset.

components	99th	95th	90th
1	82%	79%	73%
2	83%	79%	72%
3	78%	72%	71%
4	81%	75%	70%
5	81%	76%	71%
6	81%	72%	68%
7	80%	71%	79%
8	78%	74%	68%
9	75%	75%	67%
10	82%	74%	70%

Table 13: Percentage of nouns in different percentiles in the top 20 components

The second approach is to compute the mutual information of each word w to the document set D according to the following formula:

$$I(w) = p(w) \sum_{d \in D} p(x \mid d) \log \frac{p(x \mid w)}{p(x)} \qquad (7)$$

This formula was used in [94] to select the top 2000 features of a text corpus. We rank the words according to their mutual information score and compute the percentage of nouns in the top-50, top-100, and top-200 words with highest mutual information. The results given in Table 14 showed that although the nouns composed of roughly 36-39% of the words in the Binary-Multiple-500 data, they account for more than 60% of the words with highest mutual information.

Dataset	Top-50	Top-100	Top-200	All words
Binary1	64.00%	63.00%	66.50%	37.20%
Binary2	66.00%	71.00%	69.00%	37.30%
Binary3	64.00%	64.00%	65.50%	37.40%
Multi51	62.00%	61.00%	65.00%	36.50%
Multi52	70.00%	64.00%	64.00%	36.00%
Multi53	82.00%	72.00%	72.50%	37.50%
Multi101	76.00%	74.00%	73.50%	37.70%
Multi102	80.00%	78.00%	75.50%	37.80%
Multi103	66.00%	73.00%	70.50%	38.50%

Table 14: Percentage of nouns in top k words with highest mutual information

4.3.1.3 Nouns versus Random Words: Comparison

The goal of this experiment is to show that representing the documents using only nouns suffice to form clusters that are comparable to using all words. One way to show that is to cluster the documents once using only nouns and another time a random set of words drawn from the documents that is that same size as the document feature set. If the

quality of the clusters obtained using nouns are close to those using all words then this concludes that nouns are indeed a strong participant in forming the partitions. The experiment is conducted in two phases. In the first phase, we cluster the documents using the set of nouns as features. The second phase of the experiment is similar to the first one except that the features are randomly selected and includes nouns as well as non-nouns.

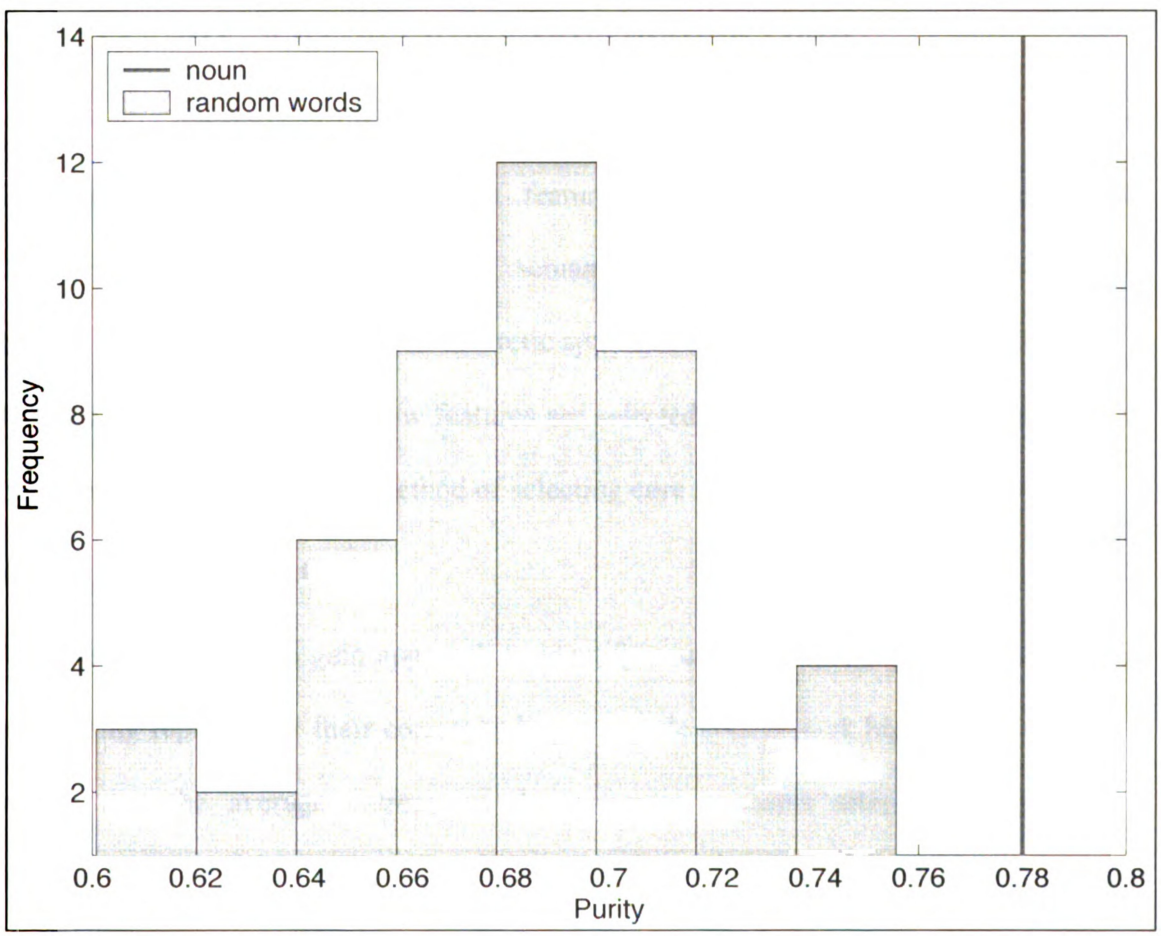


Figure 10: Purity of clusters of nouns versus random words

To be consistent with the first phase, the size of the random set of words is equal to the size of the set of nouns in order to avoid any bias related to the size of the features set. As this is a random sample, we repeat the experiment 50 times, each time using a different random sample. Min20-Max200 dataset was used in this experiment and the

results are summarized in the histogram shown in Figure 10. The vertical line shows the cluster purity sing nouns only. The purity obtained by nouns is 0.78 compared to the average purity of 0.682 using random words. These results suggest that "nouns" as a reduced feature set can produce better clusters than any other random set of words with a similar size.

4.4 Feature Selection using Information Gain

As shown in the flowchart in Figure 7 in Chapter 3, the next step after concept mapping is feature selection. We propose a feature selection approach to further reduce dimensionality and retain only core semantic features that contribute the most in clustering. We use an information theoretic approach to select salient semantic features. In Section 4.4.2 we explain how features are selected based on information gain, and in Section 4.4.2 we present our method of selecting core semantic features.

4.4.1 Information Gain

We use an information gain approach to identify nouns that have high information gain after being replaced by their corresponding semantic concepts. A high information gain means that the average entropy of the semantic concepts after disambiguation is significantly lower than the entropy of the original noun (before disambiguation. In other words, most of the documents that contain each of the disambiguated semantic concepts are from the same class (though the documents that contain the original noun belong to different classes). Unfortunately, computing information gain requires knowledge of ground truth, which is unavailable due to the unsupervised nature of the clustering task. To compensate for this, we approximate the ground truth using clusters obtained from all nouns and all semantic concepts. Since spherical kmeans is

sensitive to the initial centroids, clustering is repeated a number of iterations to obtain more stable clusters to simulate the ground truth. The following steps describe how the information gain of a noun *n* is computed:

1. Cluster the documents separately using the nouns as features and the semantic concepts as features. Let π_n be the set of noun clusters and π_s be the set of semantic clusters.

2. Compute the entropy of each noun *n* using the π_n clusters. Let *p(i|n)* be the fraction of documents containing noun *n* that belong to cluster *i*. The entropy of a noun *n* is:

$$e_{noun}(n) = -\sum_{i \in \pi_n} p(i \mid n) \log p(i \mid n) \quad (7)$$

3. Compute the average entropy of the concepts associated with this noun as retrieved by our WSD approach. Let $C(n) = \{c_1, c_2, \ldots, c_l\}$ the set of concepts that disambiguate the noun *n* and let $p(i|n, c_j)$ be the fraction of documents containing noun *n* (before disambiguation) and concept *c* (after disambiguation) that were assigned to semantic cluster j. Furthermore, $p(c_j|n)$ corresponds to the fraction of documents containing noun *n* that were disambiguated to concept c_j. The average entropy of the set of concepts associated with noun *n* is:

$$e_{concept}(n) = -\sum_{c_i \in C(n)} p(c_i \mid n) \sum_{j \in \pi_s} p(j \mid n, c_i) \log p(j \mid n, c_i) \quad (8)$$

4. The information gain from disambiguating the noun *n* is computed as the difference in entropy before and after disambiguation:

$$Gain(n) = e_{noun}(n) - e_{concept}(n) \quad (9)$$

Frequency Before Disambiguation		
	π_{n1}	π_{n2}
n	3	3

Frequency After Disambiguation		
	π_{s1}	π_{s2}
c_1	2	1
c_2	0	1
C_3	1	1

Table 15: Document distribution across clusters before and after disambiguation

Example3: Consider the example shown in Table 15 in which a noun *n* is replaced (using WSD) with {c1, c2, c3}. π_{n1} and π_{n2} are the noun clusters and π_{s1} and π_{s2} are the semantic clusters. The entropy values before and after disambiguation can be computed as follows:

$$e_{noun}(n) = -\left(\frac{3}{6}\log\left(\frac{3}{6}\right)\right) - \left(\frac{3}{6}\log\left(\frac{3}{6}\right)\right) = 1$$

$$e_{concept}(n) = \frac{3}{6}\left(-\frac{2}{3}\log\left(\frac{2}{3}\right) - \frac{1}{3}\log\left(\frac{1}{3}\right)\right) + \frac{1}{6}\left(-1 \times \log(1)\right) + \frac{2}{6}\left(-\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right)\right) = .7925$$

$$Gain\ (n) = e_{noun}(n) - e_{concept}(n) = 1 - 0.7925 = 0.2075$$

4.4.2 Core Semantic Feature Selection with Information Gain

Although existing ontology-based methods can improve document clustering, they also increase the dimensionality of the feature space. As we have noted, this can both increase computational complexity and potentially affect clustering performance due to the curse of dimensionality problem. Therefore the proposed clustering framework aims to improve document clustering using only a subset of the semantic features, the *core* features. These features are not only useful for clustering, but once identified, they may represent the main theme of the topics in the documents. In our method, we globally evaluate the significance of the nouns to decide whether their disambiguated concepts should be added to the final subset of semantic features. We establish this significance using information gain as described in Section 4.4.1.

Overall, we use two criteria to establish whether a noun is a core feature:

1. The noun should a polysemous or synonymous. We impose further restriction that it should be in the top 30% of the most frequent nouns. This final restriction further reduced the dimensionality but did not empirically change the clustering results. This is likely due to the fact that, according to our experiments, polysemous and synonymous are approximately 60% of the most frequent nouns.
2. The noun should achieve either an information gain that is greater than a predefined threshold t or an entropy equals to 0 after disambiguation, i.e., $e_{concept}(n)=0$.

Algorithm: Extracting Core Semantic Features

Input: A set of documents, D, a set of nouns N, a set of concepts C, a set of noun clusters π_n, a set of semantic clusters π_s, the list of polysemous and synonymous nouns $M_{poly,synm}$, the list of frequent noun M_{freq}, and a threshold t.

Output: Document clusters D_C using as a feature set the core semantic features F_C

1. Initialize the set of core semantic features $F_C = \varnothing$
2. for each $n \in N$
   a. Identify the concept set c associated with n using the above described method.
   b. Identify the information gain IG of n by clustering with and without the concept(s) c.
   c. if all of these conditions hold: $n \in M_{poly,synm}$, $n \in M_{freq}$ and $IG \geq t$

      then add c to the list of core semantic features i.e., $F_C \leftarrow F_C U\, c$
3. Identify the document subset $\hat{D}$, which are the documents covered by the newly identified core features F_C
4. Use spherical kmeans to cluster $\hat{D}$ with feature set F_C
5. Identify the document set $D' = D - \hat{D}$
6. Map the uncovered documents D' to the best of the existing centroids from step 4.

The first part of this requirement ensures that disambiguating an ambiguous noun makes a considerable change in the document distribution across the clusters. The second part of this requirement is related to the nouns for which the distribution of the documents across the clusters after disambiguation produce documents to produce the base clusters. However, since the set of core features is only a small subset of the entire features set,

only a subset of the documents will be clustered which we call "covered documents". That is, there will be a number of documents which

become "uncovered", not contain any of the core semantic features. We address such uncovered documents by mapping them to the clusters that have the closest centroids.

4.4.3 Experimental Evaluation

This section presents the empirical validation of our proposed algorithm for extracting core semantic features. The performance of our algorithm will be evaluated in terms of cluster purity, amount of feature reduction, and interpretability of the cluster centroids. We used Reuters2 and samples form 20newsgroups that we have discussed in Chapter 3.

4.4.3.1 Clustering using Core Semantic Features

Having described our approach, we here report on the performance of spherical kmeans clustering using core semantic features. We chose spherical kmeans as our underlying clustering algorithm for several reasons. First, it is a popular algorithm used by many ontology-driven clustering methods. Furthermore, since our goal is to evaluate the utility of the core semantic features rather than the effectiveness of the clustering algorithm, we avoid using algorithms that perform any additional feature transformations during clustering. Spherical k-means fits our criteria because it considers every feature in the input data, unlike other document clustering algorithms such as spectral clustering, which performs clustering on eigenvectors of a Laplacian matrix constructed from the feature vectors of its input data.

We compared the performance of the following four methods:

1. Spherical kmeans clustering using all nouns as features.

2. Spherical kmeans clustering using all extracted concepts (senses) from WordNet as features.

3. Spherical kmeans clustering using a combination of all nouns and their corresponding concepts (including hypernyms of each concept up to five levels distant in the WordNet concept hierarchy). This approach is similar to the one given by (Hotho et al., 2003), except we use binary features rather than the term frequency (to make an equivalent comparison with our binary weighted features).

4. Spherical kmeans clustering using the core semantic features (labeled CSF).

The clustering parameters used were the same for all four methods. The parameter k is set to the known number of classes for these data sets. Because spherical kmeans is dependent on initial centroid location, clustering is repeated 50 times for each method. Empirically, we observed that the results do not change considerably even if the number of iterations increases to 100. The quality of the clustering results is evaluated using purity, which is a supervised measure that computes the fraction of documents correctly assigned to their ground truth clusters. In addition, the amount of reduction in the number of input features is used as another criterion for evaluating the usefulness of the core semantic features. Given a baseline method B, we compute the percentage of reduction in the number of input features as follows:

$$\%\,\mathrm{Re}\,duction = \frac{\#\,Features(B) - \#\,Features(CSF)}{\#\,Features(B)}$$

where CSF denote the proposed core semantic features approach. The results of our experiments are summarized in Table 16.

The following observations are made when comparing the CSF approach against other baseline methods:

1. We achieve a feature reduction of at least 90% using the CSF approach on all the data sets (see columns 10-12 Table 16). On average, the number of core semantic features used for clustering is close to 6% of the number of nouns used for spherical k-means clustering. The ontology-driven clustering approach developed by Hotho has the highest number of features because it augments the original nouns with concepts from the WordNet hierarchy. On average, the number of features selected using our CSF approach is about 3% of the total number of features required by Hotho's method.

2. Depending on the data sets used, ontology-driven clustering may or may not improve the performance of k-means clustering using all nouns. In data sets where ontology helps (bin7, bin9, bin12, and bin13), the cluster purity obtained using core semantic features (column 9 in Table 16) is higher than that obtained using all nouns (column 3 in Table 16). For datasets where ontology does not help (bin21, bin10, multi101, and Reuters), the cluster purity of CSF is lower than that using all nouns.

Data Set	All Nouns		All Concepts		Nouns + Concepts (Hotho's method)		Core semantic features (CSF)			All Docs	Feature Reduction using CSF compared to		
	#Feat.	Purity	#Feat.	Purity	#Feat.	Purity	#Feat	Cov. D.	Cov. D. P.	Purity	All N.	All C	Hotho
bin1	3509	0.958	3790	0.959	8922	0.942	258	339	0.973	0.949	93%	93%	97%
bin2	3018	0.974	3327	0.964	7827	0.939	307	357	0.977	0.957	90%	91%	96%
bin3	4305	0.907	4630	0.917	10589	0.844	308	262	0.927	0.899	93%	93%	97%
bin4	4557	0.979	4775	0.979	10976	0.959	406	333	0.997	0.985	91%	91%	96%
bin5	3629	0.96	3907	0.962	9164	0.952	262	349	0.979	0.957	93%	93%	97%
bin6	4295	0.967	4530	0.977	10484	0.95	334	352	0.98	0.967	92%	93%	97%
bin7	4159	0.65	4451	0.917	6268	0.945	243	321	0.968	0.932	94%	95%	96%
bin8	4717	0.963	5007	0.97	11458	0.91	292	315	0.939	0.935	94%	94%	97%
bin9	4244	0.635	4569	0.917	10457	0.625	328	337	0.955	0.927	92%	93%	97%
bin10	4715	0.887	5068	0.887	11417	0.787	225	260	0.826	0.817	95%	96%	98%
bin11	4563	0.952	4843	0.973	11019	0.942	344	353	0.992	0.975	92%	93%	97%
bin12	4997	0.655	5306	0.902	12079	0.625	390	335	0.907	0.887	92%	93%	97%
bin13	4340	0.658	4559	0.883	11422	0.895	334	336	0.812	0.795	92%	93%	97%
bin14	3545	0.908	3856	0.882	8959	0.855	165	227	0.748	0.787	95%	96%	98%
multi101	5177	0.538	5563	0.458	7607	0.458	70	144	0.805	0.506*	98%	98%	99%
multiple6000	13636	0.486	14589	0.49	15569	0.352	152	839	0.559	0.439*	98%	98%	99%
Reuters2	5922	0.65	6559	0.638	8304	0.622	33	161	0.914	0.305*	99%	99%	99%

Table 1: Comparison between the core semantic feature (CSF) approach against three baseline methods (using all nouns, all concepts or Hotho's method) in terms of cluster purity and amount of feature reduction. (see discussion on how we may improve the purity of the multi-class datasets with the *). N(Nouns), C(concepts)

Overall, our proposed approach is comparable (we consider the cluster purities to be comparable if the cluster purities is less than .02) to 1 or better than clustering using all nouns in 12 of the 17 datasets, even though it has significantly less number of features.

3. The cluster purity obtained using Hotho's method tends to be closer to the minimum cluster purity obtained using all nouns or all concepts. We suspect this is due to the presence of both nouns and concepts in the feature vector, which increases the number of features considerably. In two of the data sets where ontology helped (bin9, bin12), the cluster purity is around 0.625, which is lower than the purity of using all nouns.
4. In 12 of 19 datasets, the cluster purity for our CSF approach (column 9 in Table 16) is at least comparable to the cluster purity of using all concepts even though it has less than 91% of the semantic concepts. For bin4, bin7, bin9, and multi10, the cluster purity using CSF is higher. These results suggest that the core semantic features provide a good representative subset of the ontological concepts used for clustering. There are several possible explanations for the poor performance of the CSF approach in 7 of the remaining 19 datasets. First, 3 of the datasets correspond to the case where ontology does not help (bin14, and Reuters2). Second, 2 of these datasets have a wide range of topics (Reuters2, multiple6000 and multi101), this issue is discussed in more details later in this section. Furthermore, the core semantic features are extracted using the information gain when that gain exceeds the threshold t as described in our algorithm. Threshold t is an adjustable value that determines the level of information gain sufficient to change the clustering

results when replacing a noun with its associated concept. The core semantic features can be used to both remove semantic noise introduced by the WSD process as well as exclude semantic features that have no effect on document distribution across the clusters (estimated classes). The results reported in Table 16 is based on a fixed threshold $t = 0.5$. For datasets such as bin13, we achieve a higher purity comparable to using all concepts when the threshold is reduced to 0.3.

5. In spite of the benefits gained from focusing on the core features, this small portion of the total feature set might not cover all the topics in a dataset. That is, the core semantic features might not include any features from some documents, leaving them uncovered by the features being used. This requires mapping the uncovered documents to one of the existing core feature centroids based on "closeness" of those centroids. This is especially prevalent in the case of multi-class datasets where the topics are more varied. For example, in the multiple6000 dataset the purity using all nouns is 0.486 but the purity decreases to 0.439 using only the core semantic features to cluster all the documents (uncovered as well). In the Reuters2 dataset, the purity of the all noun case, 0.65, decreased to 0.305 when using only the core semantic features to cluster all the documents.

4.4.3.2 Clusters Refinement

To improve the performance of our approach when applied to multi-class datasets, we explored different approaches for mapping the uncovered documents to the core feature centroids. We modified the mapping as follows. Instead of mapping the entire set of the uncovered documents to their core feature centroids, a random subset of the uncovered

documents were mapped to their closest centroids. After mapping, the centroids of the clusters were updated to reflect the new added documents. This was done iteratively until all the uncovered documents were mapped. This modification improved the final cluster purity for multiple6000 and Reuters. For multiple6000, the purity was raised from 0.439 to 0.495. For Reuters2, the purity of their clusters increased to 0.433 from 0.305. In the multiple6000 case, the cluster purity was comparable to those obtained using all nouns, as was the case for most of the other datasets. However, for the Reuters dataset, the purity remains below the purity of that for all nouns.

Investigating further, we noticed that the confusion matrix for Reuters had only 10 of 20 topics covered by the core semantic features, thus leaving half of the topics uncovered. However the purity of the clusters of the covered documents was high, 0.914 as shown in Table 16. Thus clustering performance was good for parts of the document set. These shortcomings might be addressed by using multiple core semantic features that cover different subsets of document set, essentially the idea of subspace clustering. Doing this iteratively would produce clusters of different parts of the document space that would have to be combined. Consider the following example. We slightly modified our approach by imposing a threshold on the distance of the documents from the closest centroid. A document is mapped to a cluster if it is within n-standard deviations distance from its centroid, otherwise, the document is not clustered. This was done to show the tradeoff between cluster purity and coverage using the core semantic features. As shown in Table 17, increasing n caused more documents to be clustered, but with lower cluster purity. This essentially means that the extracted core features cover only some of the topics from the document set. We propose to enhance the algorithm by performing

iterative core feature identification. In short, we apply the described algorithm and include uncovered documents only within some threshold *n*, a measure of distance of between a centroid and any document. Those documents that remain, that were not clustered, are redesignated as a new document set and the algorithm is reapplied. The result is a set of clusters based on different core feature sets. Again, those clusters would have to be combined at the end of the run to provide an overall result. These are suggestions we will explore in our future work.

In short, our empirical results showed that the core semantic features not only produced clusters with comparable (or sometimes higher) purity as using all concepts, it also reduces the number of features significantly. Despite its smaller number, these core semantic features are informative and sufficiently capture the main themes of a text corpus. Thus, it can be used to efficiently cluster new documents using a reduced number of semantic concepts.

Standard Deviation	# Mapped Documents	Purity of the Mapped Documents
1	161	0.936
3	162	0.93
5	236	0.757
7	598	0.521
9	1403	0.386
11	2117	0.321

Table 17: Threshold-based centroid mapping

4.4.3.3 Effect of using Core Semantic Features

Our strategy for selecting the core features using information gain can be used to find the main topics in a dataset. Generally, after clustering is performed, topics can be inferred

from the centroids of the formed clusters. If cluster purity indicates that the clusters are sufficiently representative, then we assume that the centroids of the clusters cover the main topics of the documents and each cluster contains documents that share a similar topic. In this experiment, we show the advantages of using the core semantic features for clustering. Due to space limitations, the results are shown for dataset bin10 only, although similar plots can be drawn for other datasets. First, we compare the distances of the documents to the respective cluster centroids when clustering using the core semantic features. Since there are only two clusters, the distances can be visualized in a 2-d plot as shown in Figure 11.

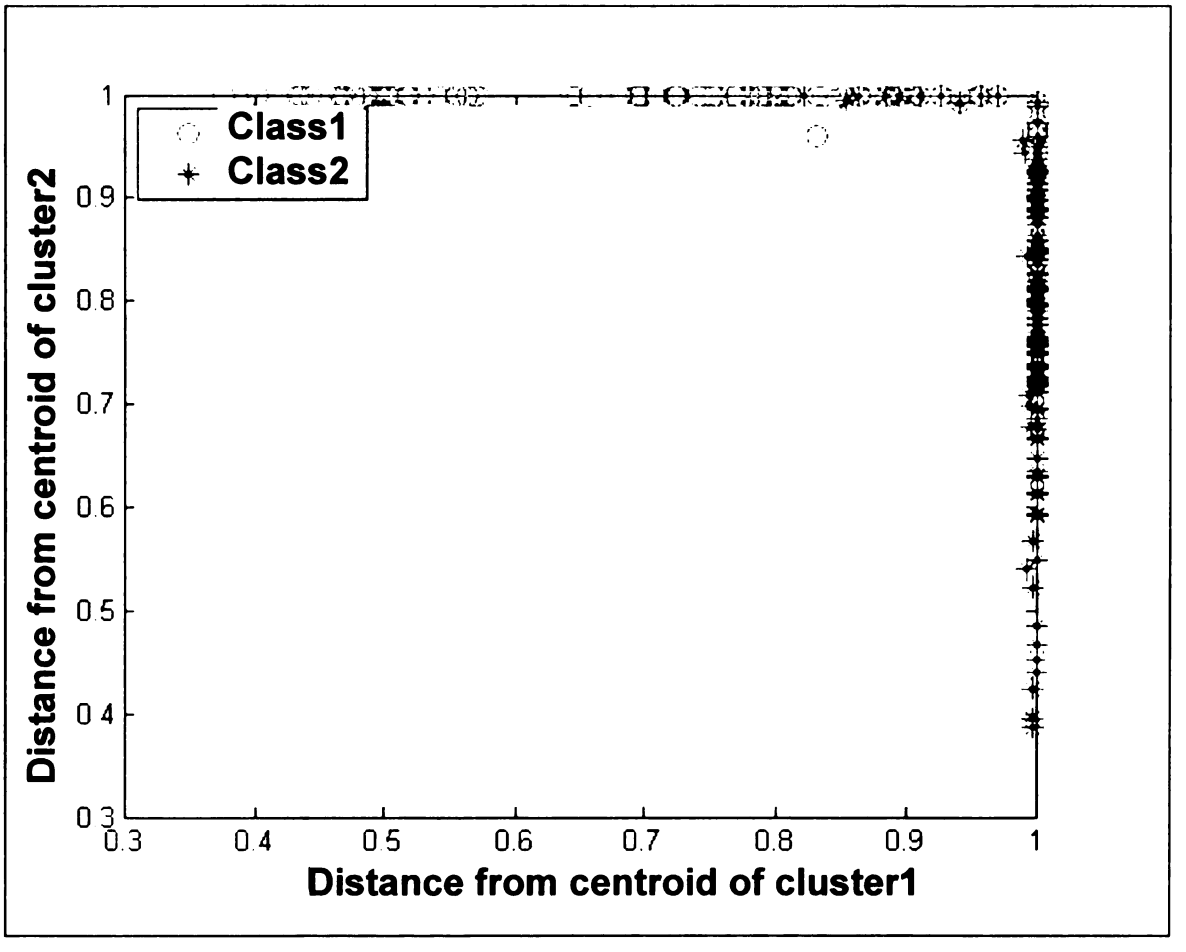


Figure 11: Distances of documents from centroids of clusters using the core concepts

Each data point is marked as either a circle or an asterisk, depending on its ground truth class. Observe that all the points marked as circles (denoting class1) have a significantly larger distance to the centroid of cluster 2 than the centroid of cluster 1. On the other hand, if we use all the concepts as features, the distance from a document to the centroid of its opposite class is no longer pushed to its maximum value (see Figure 12).

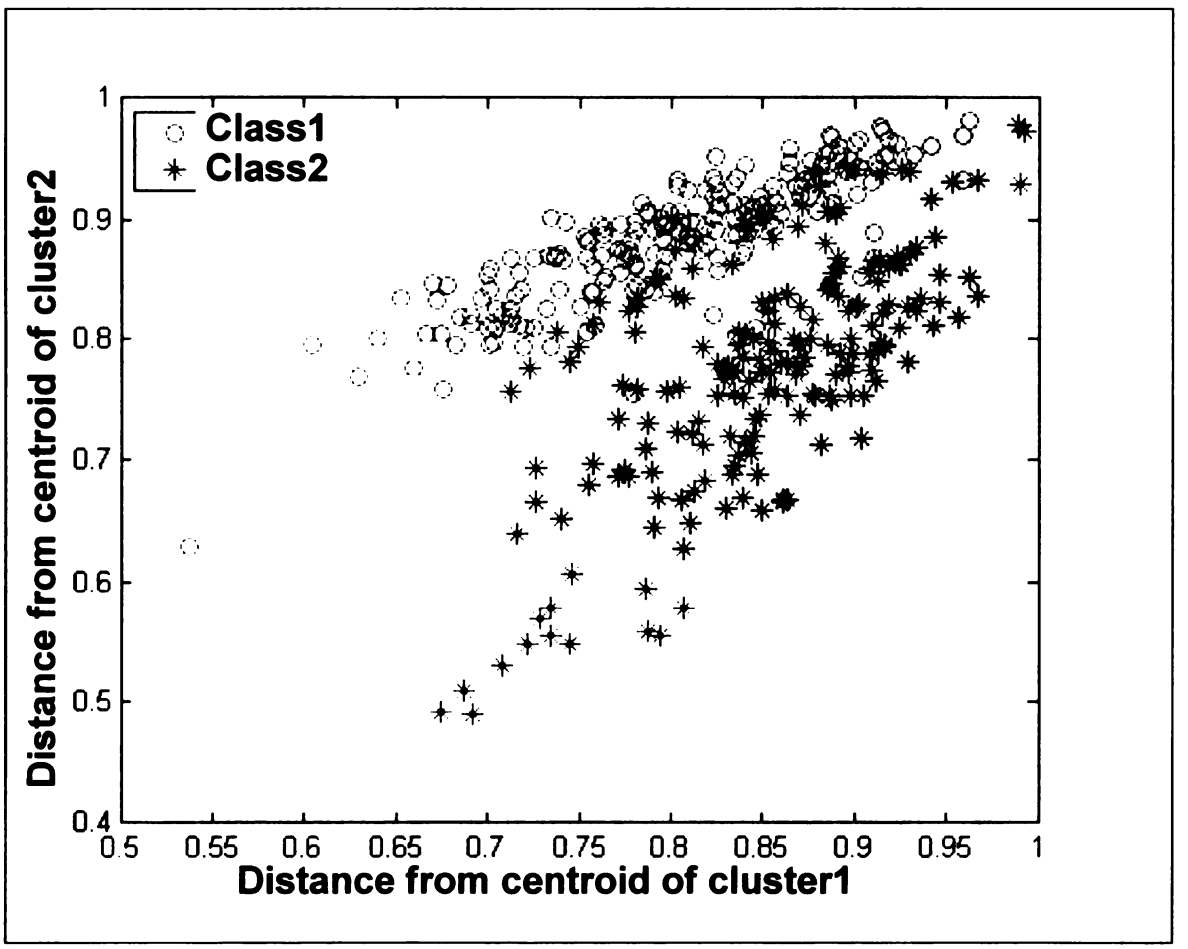


Figure 12: Distances of documents from centroids of clusters using all the concepts

As previously noted, the WSD process could potentially introduce erroneous semantic features into the text corpus. Since the core semantic features are frequent and have high information gain, we expect such features to be more accurate when used to

guide the clustering process. To examine the quality of the core semantic features, we compare the cluster centroids obtained using k-means clustering on all nouns against the centroids obtained from core semantic features. Table 18 shows the list of top 15 features that form the cluster centroids for the bin10 dataset.

Top 15 Nouns		**Top 15 Core Concepts**	
Noun centroid1	**Noun centroid2**	**Concept centroid1**	**Concept centroid2**
Space	Article	Circuit	orbit
Article	Work	Ampere	mission
NASA	Time	Transformer	satellite
System	Power	Chip	military
Orbit	Make	Police	budget
Time	Email	Resistor	landing
Make	Question	Advice	Mary
Years	Good	chips	Satellite
Earth	Circuit	wiring	Gravity
Program	Line	Connection	astronomy
Work	Problem	amplifier	exploration
Science	Id	player	Astronaut
Shuttle	Back	memory	Comet
research	Current	Port	Mars
Launch	University	Exit	Dynamics

Table 18: Top 15 features of the centroids in the clusters using nouns and the clusters using the core concepts

The dataset contains documents belonging to the "science.space" and "science.electronics" cat- egories. Using all nouns as features (columns 1 and 2 in Table 18), observe that the top features of the cluster centroids contain many noisy terms (e.g.,

make and years for centroid 1 and Article, Time, and Email for centroid 2). On the other hand, using the core semantic concepts as features (columns 3 and 4 in Table 18), most of the top features that form the centroids clearly identify the topic of the class. For example, the second cluster centroid contains concepts such as "orbit", "mission", "satellite", "atmosphere", "exploration", "gravity" and "astronomy" which all related to the "science.space" class, whereas the centroid for the first cluster includes "circuit", "ampere", "transformer", "chip" and "resistor" as its top ranked features, which is consistent with the "sci- ence.electronics" class. A question might be asked about the concept "Mary", which appears in the top 10 concepts in the "science.space". The reason is because "Mary" exists as an individual concept in WordNet and "Mary Shafer", a NASA employee, was a frequent contributor to the newsgroup "science.space". In short, the results of this section suggest that the core semantic features produce cluster centroids that are informative and relate to the main topics of the documents.

4.5 Summary

In this chapter, we presented the evaluation metrics used to validate our methods. We also describe our simplified noun-based approach baseline clustering in which the nouns are identified using WordNet and the clusters are obtained by applying spherical kmeans to the documents. Finally, we proposed a feature selection approach based on information gain in which only a subset of the semantic features is selected to be used in clustering. Our experimental results showed that the core semantic features were sufficient to not only maintain or possibly improve clustering but also substantially reduce the dimensionality of the feature set.

Chapter 5

5 Clustering Documents using WordNet: An Ensemble Approach

In this chapter we propose a new method in which we combine statistics and semantics to improve clustering. We first give a detailed description of the method; we describe how we build the noun ensemble and the sense ensemble. Then we explain how we combine the clustering solutions of both ensembles to obtain the final clusters.

5.1 Preliminaries

Given a text corpus of N documents $D = \{d_1, d_2,...,d_N\}$, our goal is to partition D into k clusters by incorporating semantic knowledge. Let $W = \{n_1, n_2,...,n_m\}$ be the corresponding set of nouns. We consider the bag of nouns representation in which each document d_i is represented as a frequency vector where each entry carries the frequency of the corresponding noun in the document. Similarly, let $V = \{s_1, s_2,....,s_r\}$ be the set of senses. Each document d_i is represented as a binary vector of 1 if the sense exists in the document and 0 otherwise.

5.2 Proposed Framework

The proposed ensemble clustering framework combines the clustering solutions obtained from the semantic similarity of the documents with those obtained based on frequency

---

Algorithm: Combined Ensemble Clustering

Input: A set of documents, D, and number of clusters k, and maximum number of iterations l

Output: Cluster membership matrix, C.

7. $\mathbf{A_f} \leftarrow$ Preprocessing(D)

8. $\mathbf{A_S} \leftarrow$ WordSenseDisambiguation($\mathbf{A_f}$)

9. Initialize: $\text{Ens}_{sense} = \varnothing$ and $\text{Ens}_{noun} = \varnothing$

10. **for** $l = 1$ **to** l **do**

    - $C_f \leftarrow$ Noun_Cluster_Membership($\mathbf{A_f}$, k)
    - $\text{Ens}_{noun} = \text{Ens}_{noun} \cup \{C_f\}$
    - $C_s \leftarrow$ Sense_Cluster_Membership($\mathbf{A_S}$, k)
    - $\text{Ens}_{sense} = \text{Ens}_{sense} \cup \{C_S\}$

11. **end**

12. $C \leftarrow$ Combine(Ens_{noun}, Ens_{sense}, k)

---

similarities. Our rationale for using ensemble clustering is that, although individual clustering solutions (using either frequency or semantic similarity) may make poor decisions regarding the cluster assignments for some documents, one may be able to improve clustering results by considering their collective decisions[2].

The proposed framework is also highly flexible because it may accommodate any baseline clustering algorithms as well as methodology for creating different instances of the ensemble. The ensemble clustering framework has two types of data inputs: (1) a document-noun frequency matrix $\mathbf{A}_f$ which corresponds to the preprocessed document vectors with frequency values, and (2) document-sense binary matrix $\mathbf{A}_S$ which corresponds to binary document vectors obtained by transforming the nouns contained in each document to their corresponding senses using WordNet. No frequency is included in the $\mathbf{A}_S$ matrix. A set of noun-based clustering solutions, Ens_{noun}, is then generated from $\mathbf{A}_f$ using the methodology presented in Section 5.3. Analogously, a set of sense-based clustering solutions, Ens_{sense}, is obtained by applying the methodology given in Section 5.4. The clustering solutions from Ens_{noun} and Ens_{sense} are then aggregated to obtain the final clustering using the approach described in Section 5.5.

5.3 Noun Cluster Membership Matrix

The nouns cluster membership matrix is generated by applying ensemble clustering to the document-noun frequency matrix $\mathbf{A}_f$. First, we randomly sample a subset of the nouns,

[2] Assuming each clustering solution is independent and is doing better than random cluster assignments.

$W_S \in W$. To alleviate bias in the sample size $|W_S|$, the number of nouns to be sampled is an integer chosen randomly between $|W_S|/2$ and $|W_S|$-1, where m is the total number of nouns in the dataset vocabulary. A truncated document-noun frequency matrix, $\mathbf{A_f}$` is then created by choosing only the nouns in each document that belongs to the subset W_S. Once the truncated matrix is obtained, the weights for each document vector are further normalized using the TFIDF method [90]. Finally, we apply the spherical k-means algorithm to obtain an $N \times k$ frequency-based cluster membership matrix, C_f, whose $(i,j)^{th}$ element is equal to 1 if the document d_i belongs to cluster j and 0 otherwise. We repeat the process to get l clustering solutions (where $l = \{10, 20, 30\}$) that will be used to compose the Noun Ensemble (Ens_{noun}) as we will describe in Section 5.5.

5.4 Sense Cluster Membership Matrix

Our approach for creating the semantic cluster membership matrix is quite similar to the approach described in the previous section. However, instead of using the document-noun frequency matrix, we use the document-sense binary matrix $\mathbf{A_S}$ as input to the ensemble clustering algorithm. We randomly sample a subset of the senses where the sample size is a random integer chosen between $|V_S|/2$ and $|W_S|$-1. A truncated document-sense binary matrix $\mathbf{A_S}$` is then created by removing all the senses not included in the sample. After normalization using TFIDF, we apply the k-means clustering algorithm to obtain the sense-based cluster membership matrix C_S, where the element $C_S(i,j)$=1 if the

document d_i belongs to cluster j and $C_S(i,j)=0$ otherwise. In addition to this approach, we have experimented with other approaches for generating the sense-based cluster membership matrix. One approach is to apply agglomerative hierarchical clustering on the Wu-Palmer similarity matrix between senses in a document. We then sample the senses based on the cluster cohesion (e.g., sample 70% of the senses in the cohesive clusters and 20% from the non-cohesive clusters). Although this approach intuitively helps to ensure that the sample retains some of the "core" senses of the document, our experimental results do not seem to show any significant improvements in the clustering results despite being more computationally intensive.

5.5 Combined Ensemble Clustering

This section describes our proposed approach for combining the Noun Ensemble (Ens_{noun}) with the Sense Ensemble (Ens_{sense}). First, an $N \times N$ weighted co-association matrix S_f is computed from the set of frequency-based cluster membership matrices that we obtained as described in section 5.6.3. The co-association matrix represents the number of times a pair of documents is assigned to the same cluster in the ensemble, weighted by the "quality" of the individual clustering solution. Formally it is iteratively computed from the frequency-based cluster membership matrices C_f as follows:

$$S_f^{(t+1)} = S_f^{(t)} + w_t C_f^{(t)} C_f^{(t)T} \qquad (1)$$

where the matrix product $C_f^{(t)} C_f^{(t)T}$ is a binary 0/1 matrix that indicates whether a pair of documents belongs to the same cluster during the t^{th} iteration of the ensemble and the weighting factor w_t measures the quality of the clustering. The matrix product $C_f^{(t)} C_f^{(t)T}$ is also known as an incidence matrix in clustering literature and will be denoted by the symbol I_t in the remainder of this section. In principle, if the true clusters are known, w_t is given by the accuracy of the individual clustering solution. However, since clustering is an unsupervised learning task, its quality needs to be estimated using unsupervised cluster validity measures such as the correlation coefficient. Given a document-noun frequency matrix $\mathbf{A_f}$, we may compute the cosine similarity matrix for all pairs of documents by normalizing the rows in $\mathbf{A_f}$ to unit length and multiplying the normalized matrix with its transpose, i.e.

$$\Sigma_{cosine} = \hat{A}_f \hat{A}_f^T \qquad (2)$$

The weighting factor w_t is then computed by correlating the cosine similarity matrix Σ_{cosine} with the incidence matrix I_t. The higher the correlation is, the greater the level of agreement between the clustering results and the document similarity matrix. Σ_{cosine} Equation (3) is iteratively updated using all the clustering solutions on the noun frequency matrix. The weighted co-association matrix S_f effectively encodes the likelihood that a pair of documents is in the same cluster based on its noun frequency

information. Furthermore, we applied k-means on S_f to obtain the final clustering for the Noun Ensemble. The procedure is repeated to obtain the co-association matrix S_S:

$$S_s^{(t+1)} = S_s^{(t)} + w_t C_s^{(t)} C_s^{(t)T} \quad (3)$$

where the incidence matrix $C_S^{(t)}$ $C_S^{(t)T}$ depends on the clustering solutions of the document-sense binary matrix A_S, and the weighting factor w_t is the correlation coefficient between the cosine similarity of documents (computed from the document-sense binary matrix A_S) and the incidence matrix. The overall Sense Ensemble can be obtained by applying the k-means algorithm to the weighted co-association matrix. However, since our intention is to combine the Ens_{noun} with Ens_{sense}, we linearly aggregate their weighted co-association matrices as follows:

$$S_{Combined} = \alpha S_s + (1-\alpha) S_f \quad (4)$$

where α is a parameter that governs the tradeoff between using both ensembles. We will apply k-means on the combined weighted co-association matrix to obtain the clusters.

5.6 Experiments

We used *Reuters2*, *Multiple-6000* and *Binary-Multiple-500 dataset* in our experiments in this chapter. The experiments below show the effect of using all senses or all nouns on clustering versus using our proposed ensemble algorithm.

5.6.1 Document Clustering using Senses

After fixing the sense of each noun using the word disambiguation approach described in chapter 3, our next step was to compare the quality of clusters obtained using the senses

as features instead of their original nouns. For Reuters2 dataset, we started with 5922 nouns as features. After applying WordNet, the data is transformed into a document-sense matrix with 6559 senses. Likewise in the Multiple-6000 dataset, we initially started with 13801 nouns that correspond to 14589 senses produced by our word sense disambiguation method.

Clearly, the number of dimensions has increased after word sense disambiguation since the algorithm may resolve the same noun into multiple senses depending on the context of the documents. Next, we apply k-means clustering on both the document-noun frequency matrix A_f and document-sense binary matrix A_S. Table 19 shows the results of the clustering, which are based on the average entropy for 50 iterations of applying k-means with different initial centroids. These results however do not seem to support the need to replace the nouns with their corresponding senses. In fact, the use of sense information seems to degrade the entropy significantly for *Reuters2* data set.

Dataset	Nouns-frequency matrix	Semantic binary matrix
Multiple-6000	0.86	0.88
Reuters2	0.97	1.19

Table 19: Comparison of entropy for K-means

One possible explanation for the poor results is related to increasing the number of dimensions when we replace the nouns by their senses. Another explanation is due to replacing some of the nouns with incorrect senses. Nevertheless, we still expect some documents to be correctly placed in the right cluster because of the word sense disambiguation. What is needed is a clustering algorithm that:

1. Better utilizes the sense information in combination with the term statistics information.
2. Deals with the increasing number of dimension when replacing the nouns by their senses.
3. Tolerant to noise when incorrect senses are used.

Because of its flexibility and robustness, we conjecture that ensemble clustering is an appropriate approach for combining term statistics with semantic knowledge for document clustering.

5.6.2 Ensemble Clustering

Ensemble clustering is a technique that was developed to improve the performance of clustering algorithms by aggregating the results from multiple runs to obtain the final clusters. In this work, we combine the clustering results from both the Sense Ensemble (denoted as Ens_{sense}) and the Noun Ensemble (denoted as Ens_{noun}) into one final clustering (denoted as Ens_{both}). Our motivation for using ensemble clustering is because of its flexibility to accommodate any input data matrix (either the document-noun frequency matrix, the document-sense binary matrix, or both), its ability to deal with high dimensionality via feature subsampling, and its resilience to noise and other variability in the data. Figure 13 shows the results of applying the three ensemble clustering methods to the *Reuters2* and *Multiple6000* datasets. The number of clustering solutions created in each ensemble varies from 10 to 30.

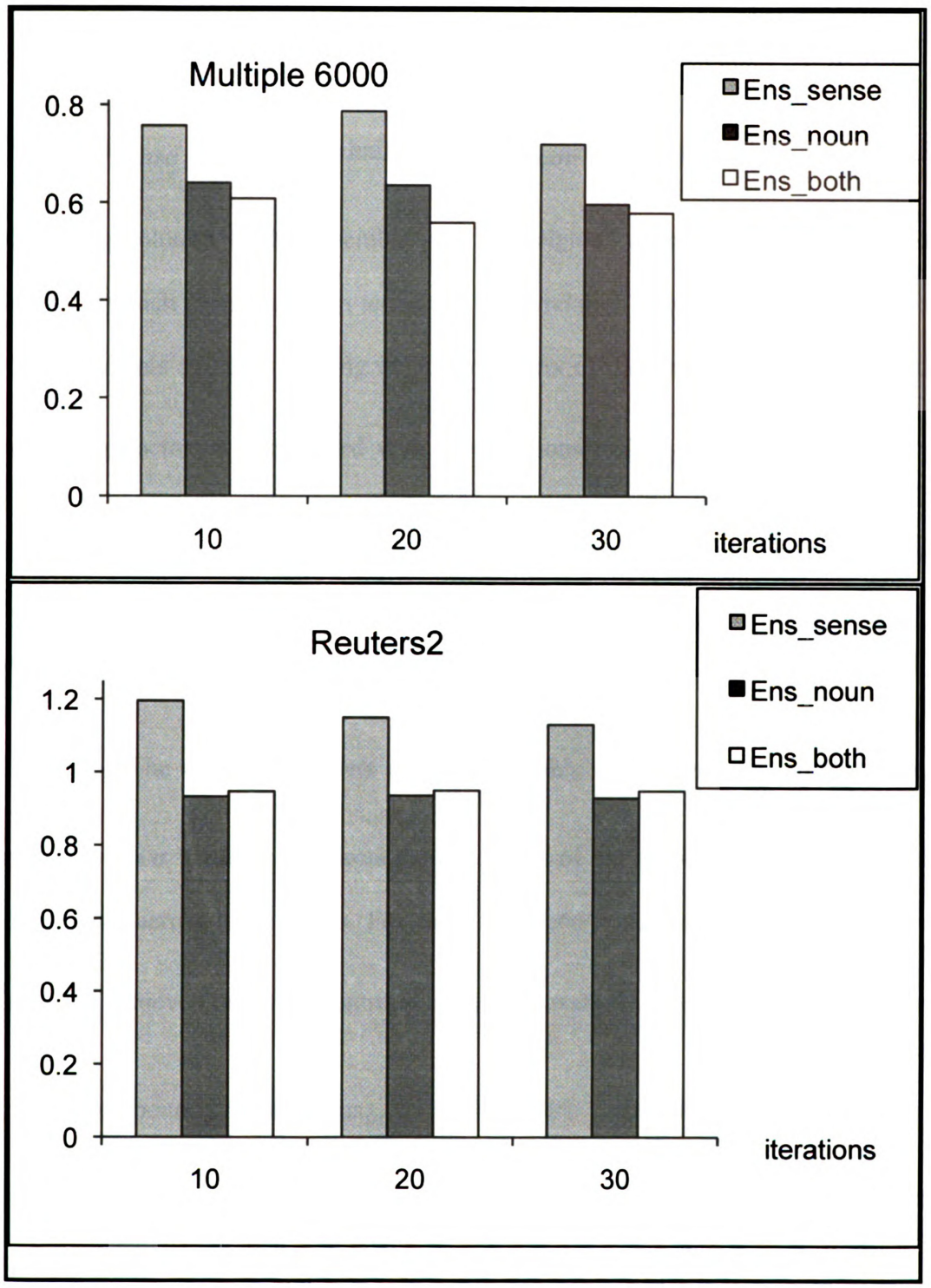


Figure 13: Comparison of entropy values of Ens_{noun}, Ens_{sense} and Ens_{both} clusters for Multiple6000 and Reuters2 datasets

To generate the final clustering Ens_{both}, we combine half of the clustering solutions from Ens_{sense} with another half from Ens_{noun}. As mentioned in Section 5.5, each clustering solution in the ensemble will be weighted according to the quality of their clusters (which is measured in terms of the correlation between the cosine similarity of the documents and the resulting incidence matrix of the clusters). We observed that the weighting factors w_t associated with the solutions in Ens_{noun} were generally greater than the weighting factors for Ens_{sense}. For example, for the *Reuters2* dataset the average of the weights of the Ens_{noun} was approximately .5 compared to .3 in the Ens_{sense}. The weighted clusters in each ensemble were aggregated in a co-association matrix (with $\alpha = 0.8$) that reflects the consensus of the individual runs on allocating the documents across the clusters. For the *Multiple6000* dataset, the compound ensemble Ens_{both} achieved the lowest entropy score. For example, after 20 runs, the entropy score for the compound ensemble Ens_{both} is .559 which is considerably lower than the scores for the Noun Ensemble .636 and the Sense Ensemble .787. This result suggests that our compound ensemble method is capable of enhancing the clustering results by taking advantage of the variability in the clustering solutions obtained from the term statistics and semantic knowledge. Even though the semantic binary model has higher entropy than noun frequency model, it still provides useful solutions that can be exploited

by our compound ensemble.

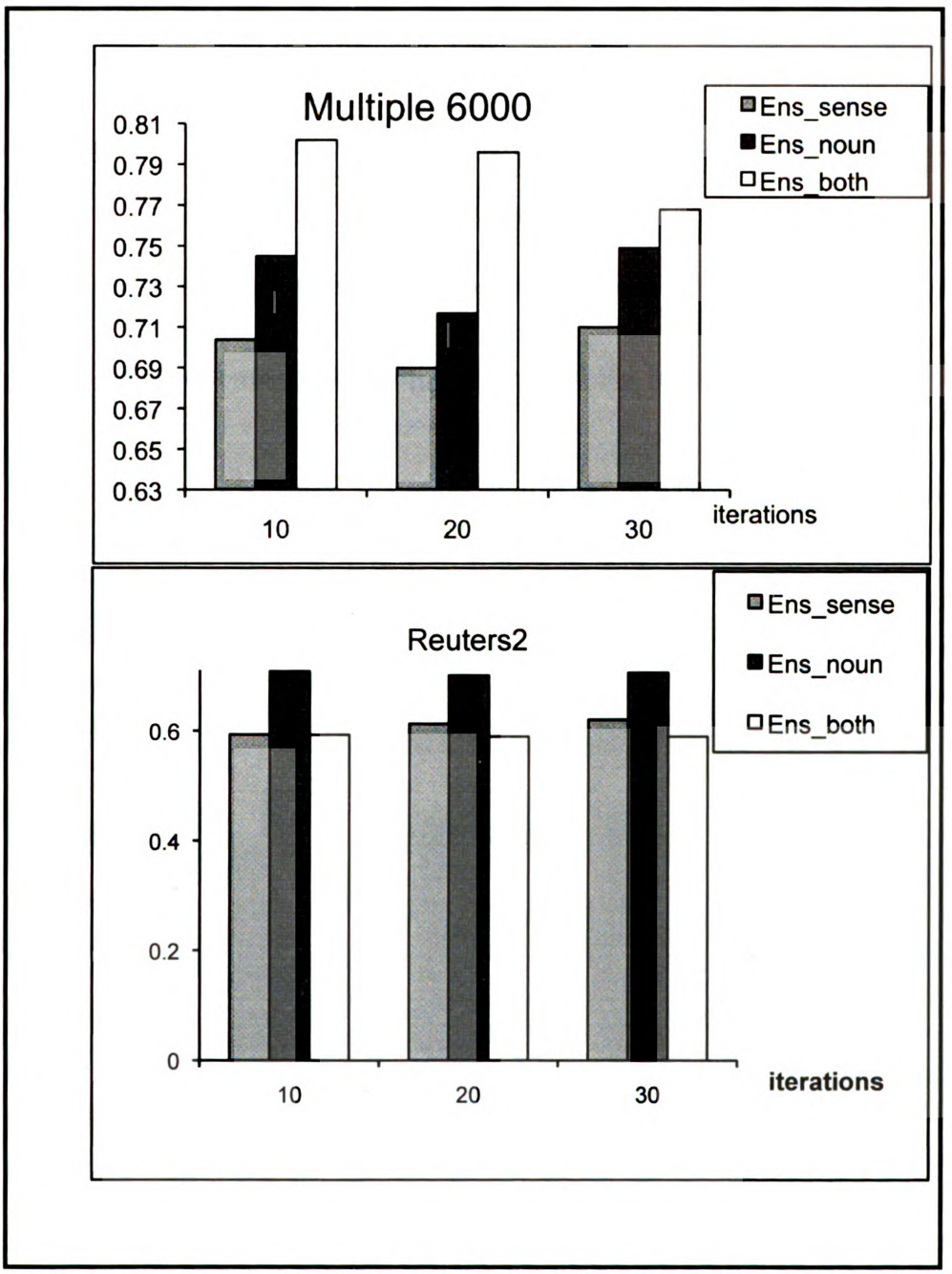


Figure 14: Comparison of purity values of Ens_{noun}, Ens_{sense} and Ens_{both} clusters for Multiple-6000 and Reuters2 datasets

Furthermore, all the ensemble clustering results are significantly better than the results for individual runs even though it uses only a sample of the original features.

In *Reuters2* dataset, no significant improvement was observed for the compound ensemble Ens_{both} over the Ens_{noun}. This result suggests that the nature of the data set also plays a significant role in determining the effectiveness of incorporating semantic knowledge from WordNet.

Figure 14 shows the purity values for the different ensemble clustering methods. Once again, the compound ensemble Ens_{both} achieved the highest purity score .802 for the *Multiple-6000* dataset whereas the noun frequency ensemble Ens_{noun} has the highest purity for the *Reuters2* dataset.

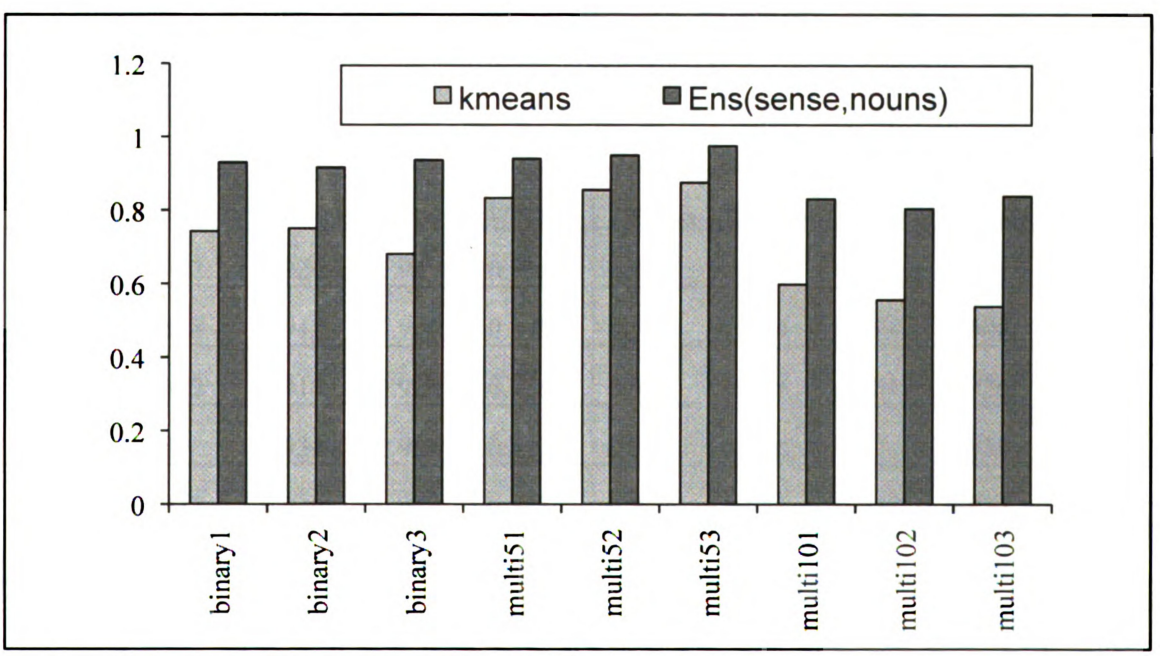


Figure 15: Comparison between k-means using cosine and the combined ensemble

Finally, it is worth noting that our proposed method using the compound ensemble clustering improved the clusters quality compared to LSI for both datasets. The lowest entropy obtained for *Reuters2* in LSI was 1.01 using 100 components compared to

.947 after 10 iterations using our compound ensemble method. For the *Multiple-6000* dataset, LSI achieved entropy of 1.13 with 100 components compared to .559 when applying the compound ensemble at 20 iterations.

Table 20 shows a comparison between the purity of the clusters obtained using spherical k-means represented by the Baseline and the corresponding purity produced by the combined ensemble Ens_{both} on the *Binary-Multiple-500* dataset. Empirical results suggest an improvement for all 9 datasets compared to the Baseline. For example, with multi101 a purity of .83 was obtained compared to .742 purity on the Baseline. Likewise, Ens_{both} exhibited an improvement for multi102 and multi103, 9% and 14% relative enhancement in the quality of the partitions was observed for both datasets, respectively.

	l=10			*l*=20			*L*=30		
sets	Ens_{sense}	Ens_{noun}	Ens_{both}	Ens_{sense}	Ens_{noun}	Ens_{both}	Ens_{sens}	Ens_{noun}	Ens_{both}
Binary1	0.922	0.924	0.93	0.916	0.928	0.928	0.916	0.92	0.924
Binary2	0.888	0.912	0.906	0.898	0.908	0.914	0.894	0.908	0.916
Binary3	0.922	0.914	0.928	0.936	0.922	0.936	0.924	0.926	0.928
multi51	0.89	0.934	0.934	0.912	0.942	0.936	0.92	0.938	0.940
multi52	0.927	0.939	0.949	0.931	0.919	0.941	0.925	0.925	0.933
multi53	0.959	0.973	0.975	0.953	0.971	0.965	0.947	0.959	0.963
multi101	0.64	0.754	0.742	0.718	0.798	0.786	0.754	0.792	0.830
multi102	0.702	0.746	0.804	0.754	0.754	0.774	0.764	0.762	0.790
multi103	0.77	0.76	0.788	0.774	0.828	0.838	0.768	0.818	0.836

Table 20: Ensemble results for nouns senses and the both combined for Binary-Multiple-500

Table 20 below shows the clusters purity for the three ensembles for all 9 datasets. In Figure 16, we plot the purity results with 30 clustering solutions.

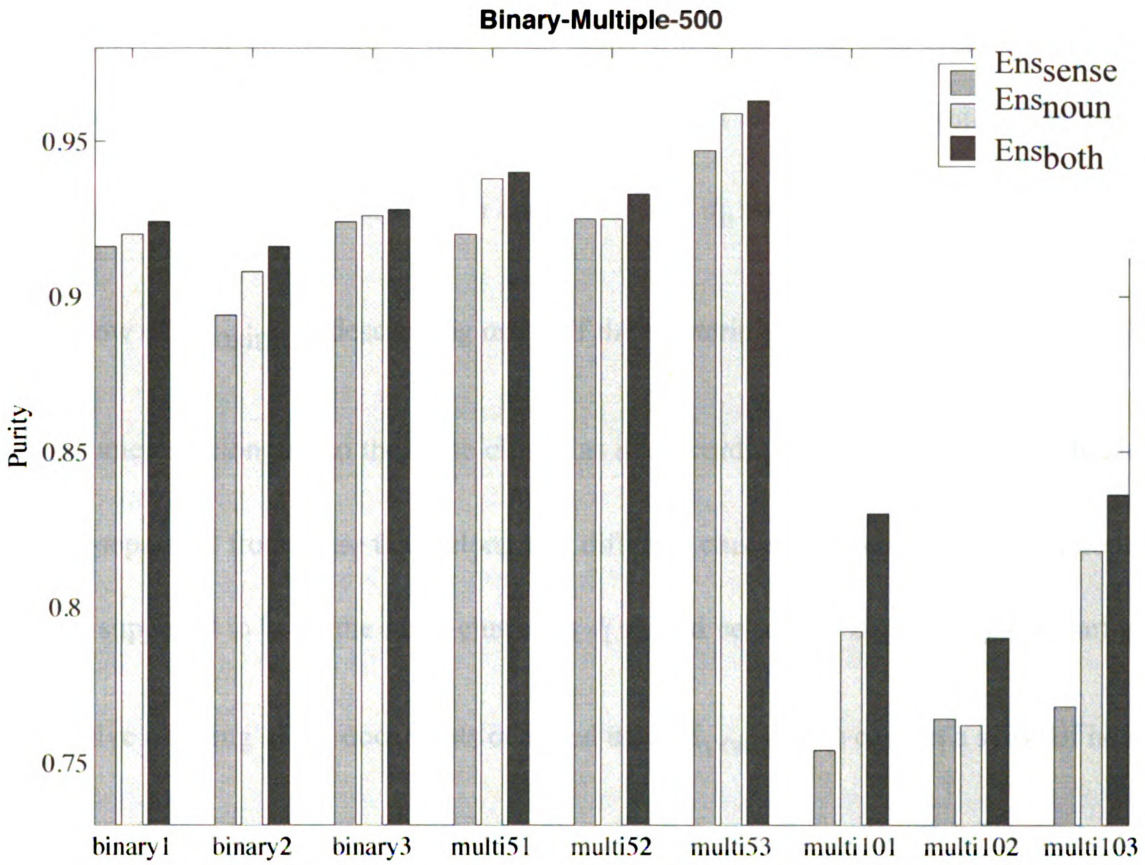


Figure 16: Ensemble results for nouns senses and both combined for Binary-Multiple-500

5.6.3 Value of Sense Information

Our experimental results suggest that incorporating sense information is helpful for clustering the *Multiple-6000* dataset but not for the *Reuters2* dataset. To validate this result, we compare the cosine similarities obtained from the document-noun frequency matrix $\mathbf{A}_f$ and the document-sense binary matrix $\mathbf{A}_S$ against the ground truth clusters. If sense is indeed helpful, we expect the similarity between documents that belong to the same cluster to increase when using the sense information as opposed to using the

frequency information. To measure the value of adding sense information, we compare the cosine similarity matrix $\mathbf{\Sigma}_{\text{cosine}}$ generated from the input data against the ground truth clusters. Our conjecture is that sense helps when the input data created from the document-sense matrix produces cosine similarities that are more "consistent" with the desired clusters. More specifically, for each document d_i, we sort the similarity values in the i^{th} row of $\mathbf{\Sigma}_{\text{cosine}}$ in descending order. If the clustering algorithm is perfect, then all the documents belonging to the same cluster as d_i (according to the ground truth) should be well-separated from those that belong to a different cluster. In other words, documents that are supposed to be in the same cluster as d_i should be ranked highest. By comparing the relative ordering of the documents obtained using $\mathbf{\Sigma}_{\text{cosine}}$, we can get a sense of how consistent is the input data with the desired clusters. We apply the Wilcoxon-Mann-Whitney (WMW) statistics [104] as our measure of consistency between the ordering of cosine similarity values and the ground truth clusters. We illustrate an example of using the statistic below.

Example 1: Let $D = \{d_1, d_2, \ldots, d_6\}$ be a collection of six documents. Suppose the cosine similarity (based on the document-noun frequency matrix) between d_1 and the remaining five documents are (0.5, 0.3, 0.7, 0.02, 0.11). Furthermore, let d_2 and d_3 be in the same cluster as d_1 (we denote them as "+" examples) whereas d_4 – d_6 are in a different cluster (we denote them as "-" examples). By sorting the documents according

to their cosine similarity to d_1, we obtain the following order: (d_4, d_2, d_3, d_6, d_5), or equivalently, (-, +, +, -. -). The rankings of the positive examples are given by the vector x = [2, 3], while the rankings of the negative examples are encoded in the vector y = [1, 4, 5]. The WMW statistic is defined as follows:

$$W = \frac{\sum_{i=0}^{n_+}\sum_{j=0}^{n_-} I(x_i, y_j)}{n_+ n_-} \tag{7}$$

where $I(x_i, y_j) = 1$ if $x_i < y_j$ and 0 otherwise, while n_+ and n_- are the number of positive and negative examples. In the previous example, W = 4/6. If all the positive examples are ranked higher than the negative examples, then W = 1. Averaging the WMW statistics for all the documents in the input data allows us to measure the consistency between the cosine similarity values obtained from the input data with the desired ground truth clusters. Table 21 shows the average WMW statistics for both *Reuters2* and *Multiple-6000* datasets when applied to the document-noun frequency and document-sense binary matrices.

	Average WMW statistics	
Dataset	Document-Noun Matrix	Document-Sense Matrix
Multiple-6000	0.857	0.86
Reuters2	0.931	0.918

Table 21: Average WMW statistics

For *Reuters2* data set, the WMW statistics becomes worse after transforming the nouns into senses. This implies that the cosine similarities obtained from the senses does

not help to improve clustering. On the other hand, the WMW statistics improves slightly for the *Multiple-6000* dataset.

5.7 Summary

This chapter presents our contribution in developing a new methodology for combining term statistics with semantic knowledge from WordNet for document clustering. Our analysis shows that a straightforward replacement of the terms with their senses may not necessarily improve the clustering results, which is consistent with some of the previous results reported in [92] and [104]. The clustering algorithm needs to be flexible and robust enough to deal with the higher dimensionality and noise due to improper selection of senses. To overcome these challenges, we propose an ensemble clustering method that systematically combines clustering solutions from the Noun Ensemble with those from the Sense Ensemble based on the consistency of their clustering solutions. Our experimental results show that the ensemble method is effective on some but not all datasets.

Chapter 6

6 Clustering Documents using Wikipedia: Concept Mapping Approach

In this chapter we investigate the benefit of incorporating Wikipedia in document clustering. We first analyze the effect of Wikipedia preprocessing and concept mapping on document clustering. We then present a method of exploiting Wikipedia's categories and use them as features. Finally, we present the effect of combining multiple sources of knowledge on document clustering.

6.1 Exploiting Wikipedia

Wikipedia has been recently used to generate a concept-based representation for a collection of documents [28][47][54]. In the context of Wikipedia, the concepts are the articles stored in Wikipedia. Like WordNet, Wikipedia also has a hierarchical relationship between the concepts/articles, though the relations are different. We will show how this different kind of ontology can be used to solve the same problem previously solved by WordNet. Some of the proposed methods that incorporates Wikipedia in document clustering, utilized the contents of the articles to find the concepts whereas others used the hyperlink network between the articles and/or the category network to compute relatedness between the concepts [47][110][50]. Content-based methods rely on finding the textual overlap between the documents and the articles

themselves whereas hyperlink-based approaches utilize the overlap with the anchor texts of the hyperlinks found within Wikipedia articles as well as the hyperlink network between the articles themselves.

In the next section, we show the results of applying the concept mapping approach described in Section 3.4 to ten datasets. We also compare our method with state of the art research that exploits and leverage Wikipedia knowledge to document clustering.

6.2 Evaluation of Wikipedia-based Concept Mapping Approach

Our method is an extension of Gabrilovich et al approach [28]. In this method, each concept in Wikipedia is given a relatedness score to a given document. The relatedness score is computed by taking the dot product between the document and the concept vectors. We modify this score by the relative frequency of mapping that concept to the document as we explained in Section 3.4.

We evaluate our method by comparing it with the noun-based baseline clustering (discussed in Section 4.3) as shown in Table 22. We noticed that for binary datasets using Wikipedia concepts as features resulted in clusters with purity as good as the Baseline for 85% of the binary datasets. However, for datasets with multiple classes, the problem becomes more challenging, therefore Wikipedia concepts were not sufficient to upgrade the clusters due to irrelevant concepts included during the concept mapping process.

We investigated this problem in more depth by examining the multi101 dataset, which has 10 classes. We measured a scatter ratio that reflects how close the classes in the space are before and after mapping the concepts. More specifically, we computed a

ratio of the scatter between classes to the scatter within classes. The between class scatter is nothing more than the average distances between the mean of each class and the centre of the data, whereas the within class scatter is the average distances of the documents within a class from their centre. If the scatter ratio is higher after mapping the concepts then this indicates that the presence of some of the mapped concepts in the feature set added to the ambiguity between the classes, thus distorted the clustering results. For multi101 dataset, the scatter score was 4.07 using only the nouns as features before mapping and 1.9 using Wikipedia. Obviously, the scatter in the dataset was imbalanced using the concepts, therefore for this dataset the concepts were not as effective.

Datasets	Baseline	Wikipedia	T_{depth4}	T_{3Up}
bin7	0.965	0.965	0.950	0.965
bin9	0.952	0.950	0.950	0.945
bin10	0.940	0.902	0.860	0.883
bin11	0.980	0.970	0.942	0.925
bin12	0.912	0.930	0.900	0.902
binary1	0.920	0.871	0.836	0.846
multi51	0.924	0.844	0.800	0.844
multi101	0.740	0.610	0.553	0.544
multiple6000	0.693	0.330	0.279	0.290
Reuters2	0.697	0.611	0.48	0.490

Table 22: Comparison between Wikipedia concepts, T_{depth4}, and T_{3Up} clusters in terms of Purity

Although our method did not produce better clusters than those obtained by the rigorous noun-based baseline, it outperformed current methods that incorporate background knowledge for document clustering. Table 23 shows the purity of the clusters

obtained when applying the different methods on one dataset. The results are the average of 10 runs of spherical kmeans. The last row in Table 23 shows the performance of our method. The other rows show the performance of the following methods: WordNet based algorithm proposed by Hotho et al [35], Hu et al's algorithm [47], a system that applies Gabrilovich et al method [28] and Huang et al approach [50] (refer to the related work chapter for description of these methods). Note that the results in the table are copied from Huang et al [50]. They experimented with the min20-max200 sample from Reuters-21578 dataset. To make an equivalent comparison with all approaches in the table, we applied our method to the same dataset.

Reference	**Purity**	**Clustering approach**
Hotho et al	.605	Bisecting kmeans
Gabrilovich et al	.607	Spherical kmeans
Hu et al	.655	Spherical kmeans
Huang et al	.670	Spherical kmeans
our method	.700	Spherical kmeans

Table 23: Comparison with other methods using min20-max200 Reuters dataset

Our proposed method outperformed all other methods and exhibited an improvement of 4.47% over the best performance reported by Huang et al. the best purity obtained by our algorithm in the experiment was 0.724.

To further investigate the capabilities of all approaches in terms of the quality of the features extracted. Table 24 presents a comparison between subsets of features extracted by these methods. The comparison is presented using document #15264 which belongs to the "copper" class. The first four rows in the table are taken from [50]. The results include some selected concepts from the feature vector without any particular

ordering where the last row in the table shows the top 12 concepts extracted by our method. All approaches did extract features related to copper mining. Hotho's WordNet based approach was not able to capture named entities existing in the original document as WordNet does not contain this information. As Hu et al [47] exploits both the category structure and the hyperlink network to extract the concepts, they have broader topics in their list of features, thus tech Cominco is expanded with Mining Companies in Canada and Con Mine [47]. Huang et al. [50] extracted their features using the anchor texts in Wikipedia. Their method did pick up features related to copper mining.

	Copper, venture, highland, valley, british, Columbia, affiliate, mining, negotiation, complete, administration, reply, silver, ounces, molybdenum
	Teck, John Towson, Cominco Arena, Allegheny Lacrosse Officials Association, Scottish Highlands, Productivity, Tumbler Ridge, British Columbia, Highland High school, Economy of Manchukuo, silver, Gold (color), Copper(color)
	Tech Cominico, British Columbia, Mining, Molybdenum, Joint Venture, Copper
	Mining, Joint Venture, Copper, Silver, Gold Ore, Management, Partnership, Product(business), Ounces, Negotiation, Molybdenum, Teck Cominco, Vice President, Consortium, Short ton.
	Copper, Highland Park New Jersey, Logan Lake British Columbia, Copper Mining in Arizona, Venture Capital, Scottish Highland Dance, North American Waterfowl Management Plan, Estella mine, Molybdenum, Codelco, El Salvador Mine, List of mines in British Columbia

Table 24: Comparing features generated by different approaches

Comparing with Gabrilovich's approach, note that our list of concepts is better than their concepts. This improvement is the result of modifying the relatedness score and ranking the concepts. For example the concept copper (color) was extracted by their method though it is not related to copper mining, but was included as a result of the textual overlap between the document and the concept.

In summary:

- Our method performed better when compared to existing approaches that incorporate external knowledge to boost clustering.
- Our concepts mapping approach was as good as the noun-based baseline when applied to binary datasets but considerably worse using multi-class datasets.

6.3 Exploiting Categories

In this section we establish an alternative set of features based on the categories from Wikipedia. The rationale is to introduce new features to the document that have high level or broader meanings as well as features with specific meanings. General categories are placed in the top levels of the Wikipedia category structure with the categories becoming more specific as you traverse down the category structure. For example, consider two documents that belong to the class "sports", d_1 is about swimming and d_2 is concerned with tennis sport. Ideally, both documents should be clustered together since they share the same topic "sports". Assume that the concept *Swimming (sports)* in Wikipedia was mapped to d_1 and the concept *tennis* was mapped to d_2. Using these concepts does not help in grouping d_1 and d_2 because they don't have concepts in common. However using the category structure, we can increase the similarity between

the documents by adding broader topic features at the top levels of the ontology. Figure 17 shows part of the categories of the two concepts *Swimming (sports)* and *Tennis.* The direct categories for *Swimming (sports)* are (swimming and Olympic sports), on the other hand (tennis and racquet sports) are the direct categories of the concept *Tennis*. Note that expanding the direct categories to a higher level connects both concepts *Swimming (sports)* and *Tennis* with the category (sports by type). Thus including this new categorical feature would increase the semantic similarity between d_1 and d_2 and increase the chances of putting them in the same cluster. This is an example that shows how categories can improve clustering.

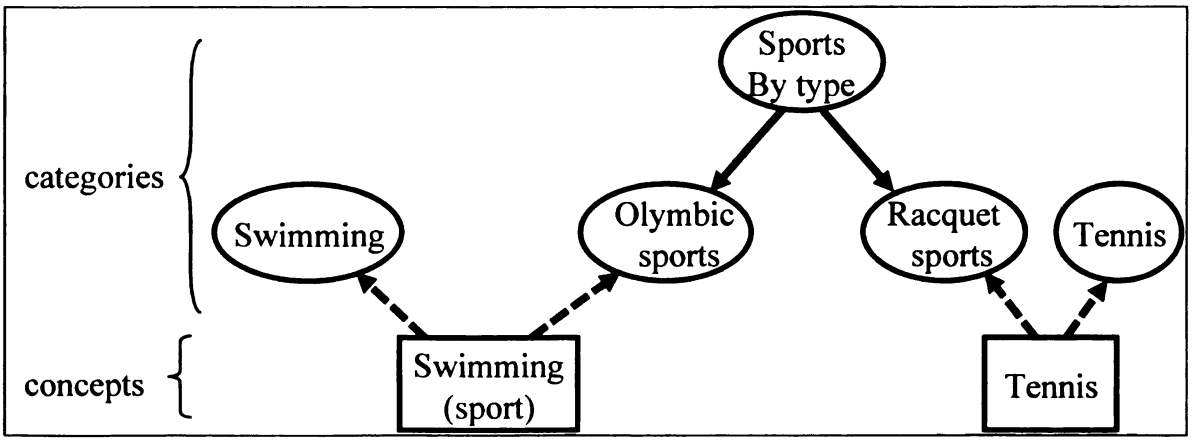


Figure 17: Part of the category structure in Wikipedia, rectangles denote concepts while circles denote categories

Therefore, we utilize the category structure from Wikipedia to extract the categories and use them as features to represent the documents.

Hu et al [49] did exploit the categories and used them as features in his work. However, they used direct categories associated with a concept and did not utilize the category structure of Wikipedia. Furthermore, their conclusion was not definitive

regarding the effectiveness of using categories "only" as features. They concluded that a weighted linear combination of the original words of the documents and the categories helps clustering. However, we argue that this analysis is incomplete and needs more investigation. That is using the direct categories of a concept (categories at the first level of the category structure) does not necessarily help clustering as we have shown in the sports example above. Furthermore, the articles are generated and edited by Wikipedia users. Thus the correctness of linking an article to a set of categories is subject to the users' opinions and knowledge. Therefore it is essential to use higher levels in the category structure, as two concepts are more likely to intersect via their corresponding categories at higher levels of the Wikipedia category structure.

In our research, we investigate these issues in addition to exploring the benefit of using categories "only" as features. Unlike Hu et al [49], we focus on using higher level categories rather than direct categories for the reasons we mentioned above. We experimented with two sets of categories: one set includes categories at a high level in the category structure with broader topics, and the other contains the direct categories and their ancestors up to three levels high in the network. The benefit of the latter option is that it maintains categories with specific meanings directly related to the contents of the article as well as categories with broader topics that might be related to the class of the original document. The discussion below explains how we obtain these sets and their characteristics.

1. Categories at depth 4 from the root (T_{depth4}). This set is obtained by traversing the category network in a Depth First Search fashion (DFS). First the direct categories of each concept are extracted $T = \{ t_j \mid t_j$ is a direct category of $c_k \}$.

Second, for all categories in the set T we extract their corresponding depth_4 categories in the category structure. The reason we choose this particular level is our intention to reduce the dimensionality that we observed using the concepts. Wikipedia maintains approximately 12,000 categories 4 levels down from the category root.

2. Direct Categories and their ancestors to 3 levels above the article's category in the category structure (T_{3Up}). We implemented the limited depth first search algorithm to get this set of categories. The depth limit is set to 3 and the algorithm starts with the direct categories of a particular concept.

The weight of each direct category t_0 (category at level 0) of a particular concept c_k is equal to the relatedness score of that concept. At level j, the weight of the category t is equal to the sum of the weights of the categories associated with its incoming-links categories at level $j-1$.

$$W_j^t = \sum_{i=1}^{|h|} W_{j-1}^t \qquad (1)$$

Here h is the number of incoming-links (children) of t.

Table 22 illustrates the results of the depth4 categories (T_{depth4}) and the 3-LevelsUp categories (T_{3up}). When compared to the noun-based baseline, the performance of both sets of categories for several datasets, especially the multi-class ones, was not encouraging. We noticed that some categories that belong to T_{depth4} had high document frequency that, in some cases, is nearly equal to the total number of

documents in a dataset. Such categories are non-discriminative and have a negative effect on clustering because their existence incorrectly increases the similarity between semantically unrelated documents. To illustrate this finding, we plotted the Cumulative Distribution Function (CDF) of the document frequency of nouns (solid curve in the figure) and the document frequency of Tdepth4 Wikipedia categories as shown in Figure 18. Recall that Reuters2 has 20 topics in total in which the maximum number of documents per topics is 200. It appears from the figure that the density of document frequency of the nouns spans the interval [0,500]. In fact F(200) =P(X<200)=0.986 which indicates that the density is mostly distributed below 200. In other words, a large portion of the nouns have document frequency less than or equal to 200. This is consistent with the fact that the datasets includes topics with mostly 200 documents. More specifically, a noun that is specifically related to a certain topic might appear in at most 200 documents (i.e. have document frequency of 200) or less. The remaining area under the curve has a value of 0.014 which represents the number of nouns with document frequency greater than 200. The existence of such features is undesirable because they cause the different topics to intertwine making partitioning more difficult. Unfortunately, in the T_{depth4} category curve in Figure 18, a large number of categories have high document frequency since the categories distribution spans a wider interval [0, 2500] than the nouns. This is at least partially responsible for the poor performance of the categories compared to the nouns. It was based on this analysis that we decided to use other category sets besides T_{depth4} which seemed to have too many common connections between document categories. We introduced T_{3Up} categories to overcome wide coverage problem observed

in the T_{depth4} categories. That is, T_{3Up} includes categories with specific topics that are semantically close to the associated concepts as well as categories with broader topics. For 6 out of 10 dataset, a slight improvement in the clusters was obtained using T_{3Up} compared to T_{depth4} while no improvement observed comparing to the noun-based baseline as shown in Table 22. We suspect that including all the categories on the path from level 0 up to level 3 in the network increases the chances of including irrelevant categories to the features set.

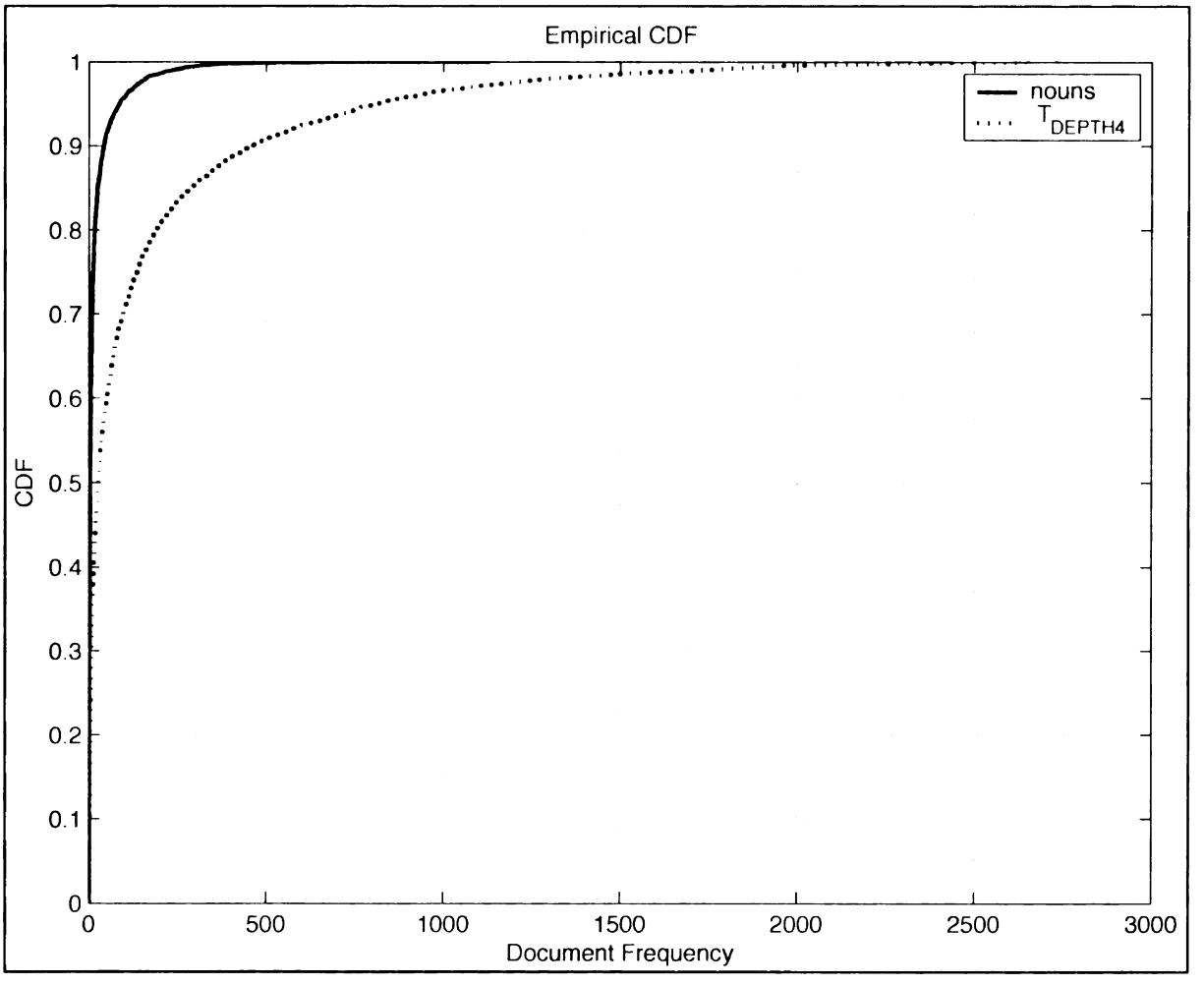


Figure 18: Cumulative distribution function of Wikipedia T_{depth4} categories and nouns for Reuters2 dataset

Especially that the category structure of Wikipedia is so highly interlinked and links doe not necessarily indicate an is-a relationship. To deal with this challenge, more research is required to better extract the categories from Wikipedia to represent the documents. The conclusion of this experiment is summarized as follows:

- Despite the extensive study that we made, both sets of categories exhibited worse performance than the concepts
- T_{depth4} categories gave comparable clusters to the nouns baseline in 50% of the datasets.
- No significance between T_{depth4} and T_{3Up} in terms of performance.

6.4 Combining Multiple Sources of Knowledge

In Chapter 6 we showed that combining the ensemble produced by aggregating clustering solutions based on the senses from WordNet and the ensemble produced based on the nouns improved document clustering. In this section, we explore the benefit of adding another ensemble based on Wikipedia concepts to the previously created ensembles. We briefly describe the three components of the new ensemble which we call Ens_{all}:

- Noun-based ensemble Ens_{noun}: this ensemble is generated by combining different clustering solutions produced by applying spherical kmeans using nouns as features.
- WordNet-based ensemble Ens_{sense}: this ensemble is generated by combining different clustering solutions produced by applying spherical kmeans using the senses extracted from WordNet. The senses were obtained by leveraging the

WordNet hierarchy as Wu-Palmer similarity measure was used to identify the most appropriate senses that disambiguate the nouns.

- Wikipedia-based ensemble $Ens_{concept}$: this ensemble is generated by combining different clustering solutions produced by applying spherical kmeans using the concepts from Wikipedia as explained in Section 3.4, these concepts are extracted based on the textual overlap between the original documents and the concepts themselves.

To build the combined ensemble, three co-association matrices have to be generated: the noun co-association matrix, the sense co-association matrix and the concept co-association matrix. In the previous chapter, Sections 5.3 and 5.4 explain how we generated the noun cluster membership matrix and the sense cluster membership matrix, respectively. Then in Section 5.5 an explanation on forming the co-association matrices for both the nouns and the senses is provided. The co-association matrix using Wikipedia concepts is generated in a similar fashion. At this point, the three co-association matrices are linearly combined according to the following equation:

$$Ens_{all} = (1-\lambda) \times Ens_{noun} + (2/3)\lambda \times Ens_{sense} + (1/3)\lambda \times Ens_{concepts} \quad (9)$$

The three ensembles are linearly combined based on the weighting factor λ. We give minimal weight to Wikipedia ensemble since it has the worst clusters in the hybrid ensemble. In the experiment, the weighting parameter λ set to 0.5, since we noticed from our experiments that it produced best results. The clustering results reported for each ensemble are based on 30 iterations. In each iteration, spherical kmeans is applied to the corresponding co-association matrix to obtain the clusters.

ENTROPY						
Dataset	**Baseline**	**Ens_{sense}**	**$Ens_{concept}$**	**Ens_{noun}**	**Ens_{both}**	**Ens_{all}**
bin11	0.090	0.077	0.204	0.107	0.103	0.079
bin12	0.273	0.302	0.313	0.254	0.263	0.222
binary1	0.270	0.289	0.394	0.272	0.270	0.262
multi51	0.345	0.290	0.556	0.249	0.228	0.217
Multiple6000	0.939	0.720	1.610	0.594	0.579	0.701
Reuters2	0.970	1.130	1.187	0.929	0.948	0.986

Table 25: Performance of the ensembles in terms of entropy

The clusters are evaluated using entropy, purity and NMI as shown in Table 25, Table 26 and Table 27, respectively.

PURITY						
Dataset	**Baseline**	**Ens_{sense}**	**$Ens_{concept}$**	**Ens_{noun}**	**Ens_{both}**	**Ens_{all}**
bin11	0.980	0.985	0.948	0.978	0.979	0.984
bin12	0.912	0.903	0.889	0.919	0.919	0.936
binary1	0.920	0.915	0.863	0.921	0.923	0.925
multi51	0.924	0.935	0.833	0.947	0.952	0.953
Multiple6000	0.693	0.710	0.493	0.749	0.768	0.721
Reuters2	0.697	0.621	0.613	0.706	0.689	0.668

Table 26: Performance of the ensembles in terms of Purity

We mainly focus on comparing the performance of Ens_{all} with Ens_{both} because our goal is to evaluate the benefit of adding Wikipedia Knowledge $Ens_{concept}$ to Ens_{both}. Empirical results showed that combining Wikipedia ensemble $Ens_{concept}$ with Ens_{both} either does not help improve clustering or adds minimal enhancement to the clusters of Ens_{both}.

NMI						
Dataset	**Baseline**	**Ens_{sense}**	**$Ens_{concept}$**	**Ens_{noun}**	**Ens_{both}**	**Ens_{all}**
bin11	0.859	0.889	0.706	0.845	0.852	0.886
bin12	0.605	0.564	0.548	0.634	0.621	0.680
binary1	0.600	0.584	0.432	0.608	0.611	0.622
multi51	0.785	0.820	0.655	0.845	0.858	0.865
multiple6000	0.509	0.520	0.291	0.610	0.612	0.563
Reuters2	0.336	0.333	0.314	0.335	0.323	0.320

Table 27: Performance of the ensembles in terms of NMI

This minimal improvement is particularly obvious using the entropy measure. Again, we explain the poor results by the poor performance of $Ens_{concept}$. More specifically, the incorrect clustering solutions that are included in the co-association matrix of $Ens_{concept}$ distorted the co-association matrix of Ens_{both} which consequently degraded the final clusters. Therefore, the individual clustering solutions of the $Ens_{concept}$ needs to be improved to effectively participate in the combined ensemble.

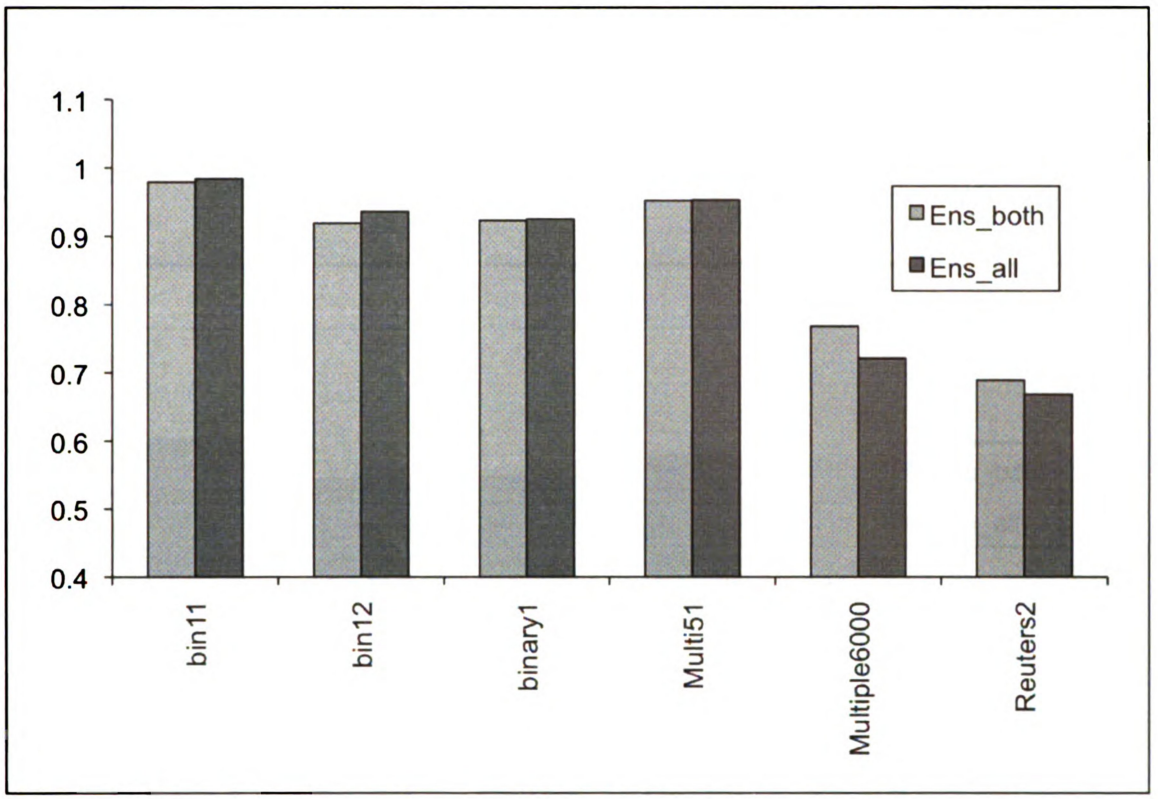


Figure 19: Comparison of the different ensembles with the Baseline in terms of Purity

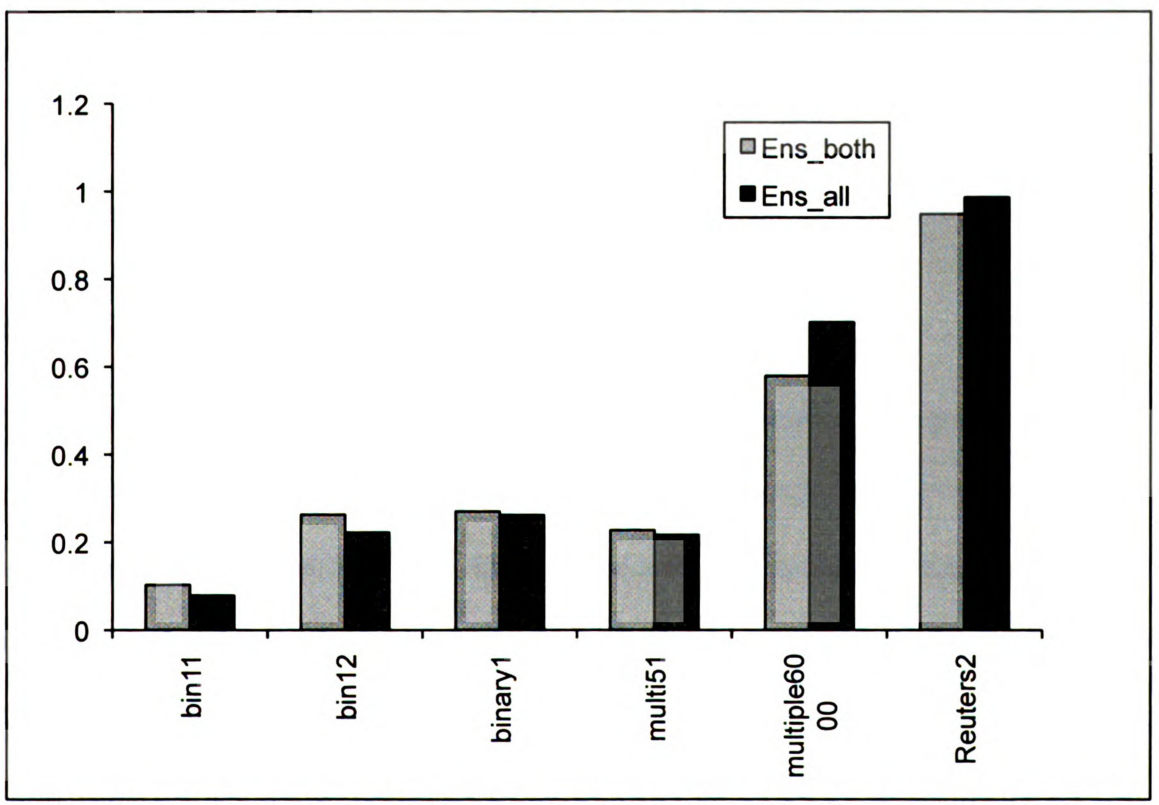


Figure 20: Comparison of the different ensembles with the Baseline in terms of Entropy

The conclusion from this experiment is that concept mapping is the key to get good clusters. Hence it remains a challenging problem that requires more research.

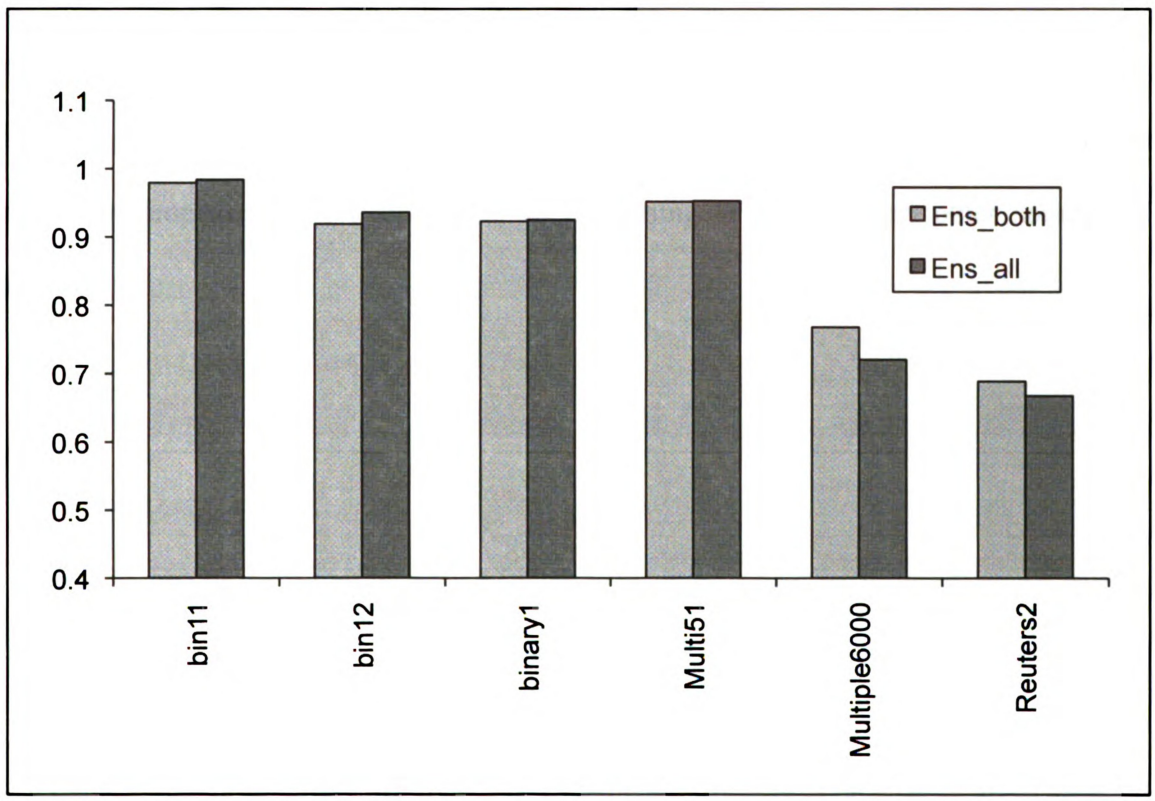


Figure 21: Comparison of the different ensembles with the Baseline in terms of NMI

6.5 Summary

In this chapter we have presented an experimental study on using Wikipedia to create a concept-based representation of a collection of textual documents. Our approach is an extension of Gabrilovich's method. Our method achieved an improvement compared to other methods when evaluated on a subset of the Reuters dataset. Additionally, we used the category graph from Wikipedia to extract categories to represent the documents. Despite the extensive experiments, the results were poor, which might be related to the

fact that Wikipedia category structure is not taxonomy with semantic relations, rather, a thematically organized thesaurus. Furthermore, we experimented with combining an ensemble of clusters using Wikipedia concepts as features with the combined ensemble of clusters using the nouns and the senses from WordNet. But because Wikipedia ensemble is not as good as the joint ensemble of nouns and senses, the results showed that integrating Wikipedia concepts did not help clustering. Hence more research is needed to improve the concepts prior to combining Wikipedia with other sources of knowledge.

Chapter 7

7 Conclusion and Future Directions

In this thesis we have investigated the problem of incorporating background knowledge into document clustering. While the idea behind this research sounds promising, its success is highly dependent on several factors:

- Effectiveness of the concept mapping approach. Specifically, a concept mapping approach must be able to address the following issues in order to provide a reasonable set of concepts for clustering:
    - Disambiguating the words in a document by mapping them to their most appropriate concepts based on the context of the document.
    - Generalizing the words in the documents by using higher level concepts of the given ontology. This step has a direct effect on increasing the similarity between documents that are semantically related yet do not have features in common.
- Effectiveness of the feature selection approach. Feature selection is needed to reduce dimensionality and eliminate erroneous concepts introduced to the feature set after concept mapping.

7.1 Contributions of this Thesis

This thesis investigates the various aspects of ontology-driven clustering research, including the strategy for concept mapping, feature selection, choice of baseline algorithm, and development of robust algorithm for combining document frequency with background knowledge from an ontology. The overall contributions of the thesis are summarized below.

7.1.1 Development of a Simplified Noun-based Approach for Baseline Clustering

A baseline clustering should reflect the best clusters that can be obtained from the data with a relatively cheap clustering algorithm. Unfortunately, the inconsistency in the baselines used for evaluation led to contradictory results about the use of an ontology in document clustering. We propose a noun-based approach from baseline clustering that is not only simple to construct but also rigorous and outperformed complicated clustering algorithm. Our analysis showed that the nouns are adequate to produce comparable or significantly better clusters than typical baseline results from published research.

7.1.2 Clustering using Core Features with High Information Gain

We proposed a methodology for clustering using the core semantic features. Our analysis shows that using all the semantic concepts may not necessarily improve the clustering results, which is consistent with some of the previous results in the literature (see Chapter 2 for details). The clustering algorithm needs to be flexible and robust enough to deal with the higher dimensionality and noise due to improper selection of concepts. To overcome these challenges, we proposed an information gain based clustering algorithm

in which only a subset of the semantic features is selected to be used in clustering. Our experimental results showed that the core semantic features were sufficient to not only maintain or possibly improve clustering but also substantially reduce the dimensionality of the feature set.

7.1.3 Hybrid Ensemble Approach for Document Clustering

We propose a hybrid ensemble approach that combines an ensemble of clusters constructed using nouns as features and an ensemble of clusters built using concepts extracted from an ontology. Our empirical results show that the compound ensemble produced better clusters than the noun-based baseline clusters.

7.1.4 Exploit Wikipedia for Document Clustering and Combine Multiple Sources of knowledge

We utilized Wikipedia to extract concept based representation for a given collection of documents. The method we used to map the concepts to the documents is based on Gabrilovich approach, however we enforce a weighting scheme on the extracted concepts to get rid of noise. We also experimented with using the categories from Wikipedia to represent the documents; nonetheless, our method needs improvement as the clustering results were not promising. Furthermore, we investigated the problem of constructing a hybrid ensemble framework using concepts from Wikipedia, senses from WordNet and the original nouns. Unfortunately, the empirical results have shown that Wikipedia did not add value to the final clusters. We suspect that this is related to the quality of the concepts extracted from Wikipedia. Having a better set of concepts from Wikipedia is expected to lead to better clustering after the combination. This will be investigated in the future work.

7.2 Future Directions

The work presented in this thesis investigates possible benefits from incorporating background knowledge into document clustering. However, this problem remains challenging and raised several questions that open different paths for future research. Next, we review some of the pending issues that we would like to investigate as part of our future work:

- Exploiting Wikipedia. Based on our previous research, the categories extracted from Wikipedia were useful for clustering binary datasets but occasionally performed good on multi class datasets. In the short term we plan to improve the mapping from the concepts to their related categories.
- Combining multiple resources of background knowledge. Previously, we combined multiple sources of knowledge via an ensemble approach in an effort to improve document clustering. However this method showed improvement for some of the datasets only. We plan to improve upon this approach by combining different types of information from the various sources. For example, WordNet has a well defined hierarchy that is reliable for computing similarities between the words in a dataset. On the other hand, Wikipedia is well-known for its wide scope of coverage. However we noticed that the concepts extracted from Wikipedia needs to be filtered to drop the noise. We plan to compute a word-word similarity matrix based on WordNet and modify the TFIDF weights of the words within concepts using this new matrix. This will give words that are semantically related within a concept higher score than loosely related ones, eventually these can be

dropped from the concept. Thereafter, we apply our concept mapping approach to extract the concept-based representation for a given dataset.

- Incorporation of semantic knowledge directly from an aexternal source of knowledge into document clustering. Current methods [104][109][110][112][115] often create a new set of semantic features that is used either solely or in combination with the original nouns in document clustering. The disadvantage of such methods is that clustering is performed independent from semantic feature extraction; which means that the effect of the erroneously generated semantic features can not be reversed at the clustering step. An alternative method to incorporate background knowledge is to design clustering algorithms that directly integrate ontology information into the clustering process.
- Predict the effectiveness of an ontology-clustering approach on several datasets. We would like to determine the usefulness of incorporating semantic knowledge from ontology in document clustering prior to applying WSD or concept mapping on a dataset. We aim to identify special characteristics of a dataset that require incorporating ontology to improve the quality of clusters. Examples of such characteristics include the percentage of polysemous and synonymous terms in a dataset or semantic relatedness of words.

APPENDIX

Top 46 Topics of Reuters-21578 Dataset

Class ID	Class	# Documents
1	Acq	100
2	alum	48
7	Bop	47
9	carcass	29
14	cocoa	59
17	coffee	100
18	copper	62
23	cotton	27
27	cpi	75
29	crude	100
33	dlr	37
36	earn	100
43	gas	33
44	gnp	100
45	gold	100
46	grain	100
50	heat	16
52	hog	16
53	housing	16
56	interest	100
58	ipi	49
59	iron-steel	51
61	jobs	50
63	lead	19
69	livestock	57
70	lumber	13
72	meal-feed	21
74	money-fx	100

Table 28: List of top 46 Topics in Reuters

Table 28(con't)

75	money-supply	100
77	nat-gas	48
82	oilseed	78
83	orange	18
89	pet-chem	29
100	reserves	51
101	retail	19
104	rubber	40
109	ship	100
111	silver	16
119	strategic-metal	19
120	sugar	100
126	tin	32
127	trade	100
130	veg-oil	93
131	wheat	21
133	wpi	24
135	zinc	20

List of Stopwords

A	d	ie	One	thats	whoever
a's	definitely	if	Ones	the	whole
able	described	ignored	Only	their	whom
about	despite	immediate	Onto	theirs	whose
above	did	in	Or	them	why
according	didn't	inasmuch	Other	themselves	will
accordingly	different	inc	Others	then	willing
across	do	indeed	otherwise	thence	wish
actually	does	indicate	Ought	there	with
after	doesn't	indicated	Our	there's	within
afterwards	doing	indicates	Ours	thereafter	without
again	don't	inner	Ourselves	thereby	won't
against	done	insofar	Out	therefore	wonder
ain't	down	instead	Outside	therein	would
all	downwards	into	over	theres	would
allow	during	inward	overall	thereupon	wouldn't
allows	e	is	own	these	x
almost	each	isn't	p	they	y
alone	edu	it	particular	they'd	yes
along	eg	it'd	particularly	they'll	yet
already	eight	it'll	per	they're	you
also	either	it's	perhaps	they've	you'd
although	else	its	placed	think	you'll
always	elsewhere	itself	please	third	you're
am	enough	j	plus	this	you've
among	entirely	just	possible	thorough	your
amongst	especially	k	presumably	thoroughly	yours

Table 29: List of Stopwords

Table 29 (con't)

an	et	keep	probably	those	yourself
and	etc	keeps	provides	though	yourselves
another	even	kept	q	three	z
any	ever	know	que	through	zero
anybody	every	knows	quite	throughout	january
anyhow	everybody	known	qv	thru	jan
anyone	everyone	l	r	thus	february
anything	everything	last	rather	to	feb
anyway	everywhere	lately	rd	together	march
anyways	ex	later	re	too	mar
anywhere	exactly	latter	really	took	april
apart	example	latterly	reasonably	toward	apr
appear	except	least	regarding	towards	may
appreciate	f	less	regardless	tried	june
appropriate	far	lest	regards	tries	july
are	few	let	relatively	truly	august
aren't	fifth	let's	respectively	try	september
around	first	like	right	trying	sep
as	five	liked	s	twice	october
aside	followed	likely	said	two	oct
ask	following	little	same	u	november
asking	follows	look	saw	un	nov
associated	for	looking	say	under	december
at	former	looks	saying	unfortunately	dec
available	formerly	ltd	says	unless	one
away	forth	m	second	unlikely	two
awfully	four	mainly	secondly	until	tow
b	from	many	see	unto	three
be	further	may	seeing	up	four

Table 29 (con't)

became	furthermore	maybe	seem	upon	five
because	g	me	seemed	us	six
become	get	mean	seeming	use	seven
becomes	gets	meanwhile	seems	used	eight
becoming	getting	merely	seen	useful	nine
been	given	might	self	uses	ten
before	gives	more	selves	using	1
beforehand	go	moreover	sensible	usually	2
behind	goes	most	sent	uucp	3
being	going	mostly	serious	v	4
believe	gone	much	seriously	value	5
below	got	must	seven	various	6
beside	gotten	my	several	very	7
besides	greetings	myself	shall	via	8
best	h	n	she	viz	9
better	had	name	should	vs	10
between	hadn't	namely	shouldn't	w	periodic
beyond	happens	nd	since	want	period
both	hardly	near	six	wants	billion
brief	has	nearly	so	was	million
but	hasn't	necessary	some	wasn't	thousand
by	have	need	somebody	way	thousands
c	haven't	needs	somehow	we	millions
c'mon	having	neither	someone	we'd	billions
c's	he	never	something	we'll	day
came	he's	nevertheless	sometime	we're	month
can	hello	new	sometimes	we've	year
can't	help	next	somewhat	welcome	yearly
cannot	hence	nine	somewhere	well	week

Table 29 (con't)

cant	her	no	soon	went	weekly
cause	here	nobody	sorry	were	saturday
causes	here's	non	specified	weren't	sunday
certain	hereafter	none	specify	what	monday
certainly	hereby	noone	specifying	what's	tuesday
changes	herein	nor	still	whatever	wednesday
clearly	hereupon	normally	sub	when	thursday
co	hers	not	such	whence	friday
com	herself	nothing	sup	whenever	i
come	hi	novel	sure	where	ii
comes	him	now	t	where's	iv
concerning	himself	nowhere	t's	whereafter	vi
consequently	his	o	take	whereas	iii
consider	hither	obviously	taken	whereby	v
considering	hopefully	of	tell	wherein	www
contain	how	off	tends	whereupon	edu
containing	howbeit	often	th	wherever	com
contains	however	oh	than	whether	org
corresponding	i	ok	thank	which	gmt
could	i'd	okay	thanks	while	text
couldn't	i'll	old	thanx	whither	html
course	i'm	on	that	who	info
currently	i've	once	that's	who's	cs

REFERENCES

[1] Banerjee S, Ramanathan K. and Gupta A. Clustering Short Texts using Wikipedia. In Proceedings of SIGIR, 2007, 787-788.

[2] Basu S., Banerjee A., and Mooney R. J., Semi-supervised clustering by seeding. In *Proceedings of the International Conference on Machine Learning 2002*, 27-34.

[3] Beil. F, Ester M, Xu X, Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD*, 2002, 436-442.

[4] BelMufti G., Bertrand P., and ElMoubarki L., Determining the Number of Groups from Measures of Cluster Validity, In *Proceedings of International Symposium on Applied Stochastic Models and Data Analysis*, pp. 404-414, 2005.

[5] Berkhin P. A survey of clustering Data mining techniques. Springer Berlin Heidelberg, 2006, 25-71

[6] Blei David, NG Andew, Jordan Michael. Latent dirichlet al.location.

[7] Bradley P., Bennett K., and Demiriz A., Constrained k-means clustering. *Microsoft Research Technical Report*,MSR-TR-2000-65, 2000.

[8] Bruijn J., Pearce D., Polleres A., and Valverde A., A semantical framework for hybrid knowledge bases. In *Knowledge and Information Systems KAIS Journal*, 2010.

[9] Cheng, Y. and Church, G.M., Biclustering of expression data, In Proceedings of the International Conference of Intelligent Systems for Molecular Biology, pp 93-103, 2000.

[10] Chernov, S. Iofciu, T. Nejdl, W. and Zhou X., Extracting semantic relationships between Wikipedia categories, In Proceedings of Workshop on Semantic Wikis, 2006.

[11] Cucerzan, S., Large-scale named entity disambiguation based on Wikipedia data, In *Proceedings of EMNLP-CoNLL*, pp 708-716, 2007.

[12] Dai W., Xue G., Yang Q. and Yu Y. Co-clustering based classification for out-of-domain documents. In *Proceedings of KDD*, pp 210-219 , 2007.

[13] Davidov D., Gabrilovich E. and Markovitch, Parameterized Generation of Labeled Datasets for Text Categorization Based on a Hierarchical Directory. In *Proceedings of the 27th annual International SIGIR conference*, pp 250-257, 2004.

[14] Deerwester S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41, pp 391-407, 1999.

[15] Demiriz A., Bennett K., and Embrechts M., Semi-supervised clustering using genetic algorithm 1999.[Online]. Available:citeseer.ist.psu.edu/demiriz99 emisupervised.htm

[16] Dudoit S. and Fridlyand J., Bagging to Improve the Accuracy of a Clustering Procedure, In Proceedings of *Bioinformatics,* vol. 19, no. 9, pp. 1090-1099, 2003.

[17] Dhillon I., Mallela S., Modha D., Information-Theoretic Co-Clustering. In Proceedings *of the International Conference on Knowledge Discovery and Data Mining*, pp 89-98, 2003.

[18] El-Yaniv R., Souroujon O., Iterative Double Clustering for Unsupervised and Semi-Supervised Learning. In *Proceedings of ECML*, pp 121-132, 2001.

[19] Fazzinga B., Flesca S., and Tagarelli A., Evaluation of two heuristic approaches to solve the ontology meta-matching problem. . In *Knowledge and Information Systems KAIS Journal*, 2010.

[20] Fern X.Z. and Brodley C.E., Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach, In *Proceedings of 20th International Conference on Machine Learning,* pp. 186-193, 2003.

[21] Fischer B. and Buhmann J.M. Bagging for Path-Based Clustering, In *Proceedings of IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 11, pp. 1411-1415, 2003.

[22] Francisco V., Gervas P., and Peinado F., Ontological reasoning for improving the treatment of emotions in text, In *Knowledge and Information Systems KAIS Journal*, 2010.

[23] Fred A., Finding Consistent Clusters in Data Partitions. In *Proceedings of the Second International Workshop Multiple Classifier Systems,* 2001.

[24] Fred A. and Jain A.K., Combining Multiple Clustering Using Evidence Accumulation, In Proceedings of *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 835-850, 2005.

[25] Fred A. and Jain A.K., Data Clustering Using Evidence Accumulation. In *Proceedings of the 16th International Conference on Pattern Recognition,* pp. 276-280, 2002.

[26] Freidman N., Mosenzon O., Slonim N. and Tishby N., Multivariate Information Bottleneck. In *Proceedings of 17th conference of UAI,* 2001.

[27] Fogarolli, A., Word Sense Disambiguation Based on Wikipedia Link Structure, In *Proceedings of the International Conference on Semantic Computing,* pp 77-82, 2009.

[28] Gabrilovich E. and Markovitch S. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *proceedings Of the 20th international Joint Conference on Artificial Intellegence (IJCAI)* 2007.

[29] Greene D., Tsymbal A., Bolshakova N., and Cunningham P., Ensemble Clustering in Medical Diagnostics, *Technical Report TCD-CS-2004-12*, Dept. of Computer Science, Trinity College, Dublin, Ireland, 2004.

[30] Goe J. , Tan P. N., and Cheng H., Semi-supervised clustering with partial background information. In *Proceedings of SIAM International Conf on Data Mining*, Bethesda, MD 2006.

[31] Gomes P., Pereira F.C., Paiva P., Seco N., Carreiro P., Ferreira J., and Bento C. Noun Sense Disambiguation with WordNet for Software Design Retrieval. In *Proceedings of the Sixteenth Canadian Conference on Artificial Intelligence*, pp 537-543, , 2003.

[32] Govaert G. Simultaneous clustering of rows and columns. Control and Cybernetics, pp 437-458, 1995.

[33] Hadjitodorov S.T., Kuncheva L.I., and Todorova L.P., Moderate Diversity for Better Cluster Ensembles, *Information Fusion,* 2005.

[34] Hofmann T. Probabilistic latent semantic indexing. In proceedings of the Twenty-Second Annual International SIGIR conference, 1999.

[35] Hotho A., Staab S., Stumme G, WordNet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, pp 541-544, 2003.

[36] http://www.daviddlewis.com/resources/testcollections/reuters21578

[37] http://www.dmoz.org/

[38] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

[39] http://lucene.apache.org/

[40] http://people.csail.mit.edu/jrennie/20Newsgroups

[41] http://www.nlm.nih.gov/mesh

[42] http://www.webconfs.com/stop-words.php

[43] http://wikipedia.edu/

[44] http://wn-similarity.sourceforge.net/

[45] http://wordnet.princeton.edu

[46] Hu B., WiKi'mantics: interpreting ontologies with WikipediA. In *Knowledge and Information Systems KAIS Journal*, 2010.

[47] Hu J., Fang L, Cao Y. , Enhancing Text Clustering by Leveraging Wikipedia Semantics. In *Proceedings of Special interest Group for Information Retrieval SIGIR* , 2008.

[48] Hu X. and Yoo I., Cluster Ensemble and Its Applications in Gene Expression Analysis, In *Proceedings of Second Asia-Pacific Bioinformatics Conference,* 2004.

[49] Hu X., Zhang X., Lu C., Park E. and Zhou X. Exploiting Wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD*

international conference on Knowledge discovery and data mining, pp 389-396, 2009.

[50] Huang A. Milne D., Frank E., and Witten I. Clustering documents using a Wikipedia-based concept representation. In *Proceedings of Advances in Knowledge Discovery and Data mining,* pp 628-636, 2009.

[51] Huang A. Milne D., Frank E., and Witten I. Clustering documents with active learning using Wikipedia. In *Proceedings of* International Conference on Data Mining, pp 839-844, 2008.

[52] Jalali V., Reza M. and Borujerdi. Information retrieval with concept-based pseudo-relevance feedback in MEDLINE. In *Knowledge and Information Systems Journal.* 2010.

[53] Jain A.K. and Dubes R.C., Algorithms for Clustering Data. Prentice Hall, 1988.

[54] Jiang J.J. and Conrath D.W., Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the international Conference on Research in Computational Linguistics.* 1998.

[55] Jing L., Ng M. K., and Huang J. Z., Knowledge-based vector space model for text clustering. . In *Knowledge and Information Systems KAIS Journal,* 2010.

[56] Jing L., Zhou L., Ng M. K. and Huang J. Z., Ontology-based distance measure for text clustering. In *Proceedings of International Conference for Data Mining SDM* 2006.

[57] Knappe R., Bulskov H. and Andreason T., .Perspectives on Ontology-Based Querying. International Journal of Intelligent Systems. 2004.

[58] Kuncheva L.I. and Hadjitodorov S.T., Using Diversity in Cluster Ensembles, In *Proceedings of IEEE International Conference Systems, Man, and Cybernetics,* 2004.

[59] Lang K., NewsWeeder: Learning to Filter Netnews. In *Proceedings of IEEE Conference on Machine Learning,* pp 331-339, 1995.

[60] Lange T., Roth V., Braun M.L., and Buhmann J.M., Stability-Based Validation of Clustering Solutions, In *Proceedings of Neural Computation,* vol. 16, pp. 1299-1323, 2004.

[61] Law M.H. and Jain A.K., Cluster Validity by Boostrapping Partitions, Technical Report MSU-CSE-03-5, Michigan State University, 2003.

[62] Law M. H. C., Topchy A., and Jain A. K. Clustering with Soft and Group Constraints. In *Proceedings of the Joint IAPR Workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition*, pp 662-670, 2004.

[63] Levine E. and Domany E., Resampling Method for Unsupervised Estimation of Cluster Validity, In *Proceedings of Neural Computation,* vol. 13, pp.2573-2593, 2001.

[64] Lewis D. Reuters-21578 text categorization test collection, 1997

[65] Li Y., Luke R.W.P. , Ho E.K.S. and Chung F.L. Improving Weak Ad-Hoc Queries using Wikipedia as External Corpus. In *Proceedings of Special Interst Group for Information Retrieval SIGIR*, pp 797-798, 2007.

[66] Lin, D. An information-theoretic definition of similarity. In *Proceedings of the international Conference on Machine Learning.* 1998.

[67] Lu Z. and Leen T., Semi-supervised learning with penalized probabilistic clustering. In *Advances in Neural Information Processing Systems (17).*

[68] Madeira, S.C. and Oliveira, A.L. Biclustering algorithms for biological data analysis: a survey, In *Proceedings of the IEEE Transactions on computational Biology and Bioinformatics*, pp 24-45, 2004

[69] Mann H. B., Whitney D. R. On a test whether one of two random variables is stochastically larger than the other. *Annals of Mathmatical Statistics*, 18, pp 50-60, 1947.

[70] Maulik U. and Bandyopadhyay S., Performance Evaluation of Some Clustering Algorithms and Validity Indices, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1650-1654, 2002.

[71] Milne D., Computing Semantic Relatedness using Wikipedia Link Structure. In Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRSC) 2007

[72] Milne D, Medelyan O. and Witten H. I., Mining Domain Specific Thesauri from Wikipedia: A case study. In Proceedings of the International Conference on Web Intelligence (IEE/WIC/ACM WI'2006) Hong Kong, 2006.

[73] Milne D, Medelyan O. and Witten H. I. An Effective, low cost measure of semantic relatedness obtained from Wikipedia Links. *In the AAAI Workshop on Wikipedia and Artificial Intelligence,* 2008.

[74] Milne D and Witten H. I., Learning to link with Wikipedia. In Proceedings *of International Conference on Information and Knowledge Management CIKM, New York*, pp 509-518, 2008.

[75] Mihalcea, R., Using Wikipedia for automatic word sense disambiguation, In Proceedings of NAACL HLT, 2007.

[76] Miller J., WordNet: a lexical database for English, Communications of the ACM, pp 39-41, 1995.

[77] Minaei B., Topchy A., and Punch W., "Ensembles of Partitions via Data Resampling," *Proc. Int'l Conf. Information Technology: Coding and Computing,* 2004.

[78] Monti S., Tamayo P., Mesirov J., and Golub T., Consensus Clustering: A Resampling Based Method for Class Discovery and Visualization of Gene Expression Microarray Data, In Proceedings of *Machine Learning Journal,* vol. 52, pp. 91-118, 2003.

[79] Moravec P., Kolovrat M. and Snasel V., LSI vs. WordNet ontology in dimension reduction and information retrieval. *Proceedings of the Dateso 2004 Annual International Workshop on Databases, Texts, Specifications and Objects*, 2004.

[80] Natural Language ToolKit. http://www.nltk.org/Home

[81] Ni X., Quan X., Lu Z., Wenyin L. and Hua B., Short text clustering by finding core terms. In *Knowledge and Information Systems KAIS Journal*, 2010.

[82] Phan, X., Nguyen L. and Horiguchi, S. Learning to Classify Short and Sparse Test & Web with Hidden Topics from Large-scale Data collection, WWW, pp 21-25, 2008.

[83] Pedersen T., Patwardhan S., and Michelizzi J. WordNet Similarity–Measuring the Relatedness of Concepts. *In Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 2004, 1024-1025.

[84] Porter M.F., An algorithm for suffix stripping, *In Proceedings of the Program journal*,14(3) pp 130–137, 1980.

[85] Rand W.M., Objective Criteria for the Evaluation of Clustering Methods, *J. Am. Statistical Assocociation,* vol. 66, pp. 846-850, 1971.

[86] Recupero D. A new unsupervised method for Document Clustering by using WordNet Lexical and Conceptual Relations. In *Proceedings of Information Retrieval,* pp 563-579, 2007.

[87] Resnik O., Semantic Similarity in a taxonomy: An Information-Based Measure and its application to Problems of Ambiguity and Natural Language. Journal of Artificial Intelligence Research, pp 95-130, 1999.

[88] Rosen-Zvi M, Griffiths T., Steyvers T., Smyth P., The author-topic model for authors and documents, In *proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pp 487-494, 2004.

[89] Roth V., Lange T., Braun M., and Buhmann J., A Resampling Approach to Cluster Validation. In *Proceedings of the Conference of Computational Statistics,* pp. 123-128, 2002.

[90] Salton G. and Buckley C. Term-weighting approaches in automatic text retrieval. In *Proceedings of Information Processing & Management.* 24 (5), pp 513–523, 1988.

[91] Salton G., Wong A., Yang C. S., A vector space model for automatic indexing, In *proceedings of Communications of the ACM*, pp 613-620, 1975.

[92] Sedding J., Kazakov D., WordNet-based text document clustering. In *Proceedings of the 3rd workshop on Robust Methods in Analysis of Natural Language Processing Data*, pp 104-113, 2004 ,

[93] Slonim N., Friedman N. and Tishby N., Agglomerative Multivariate Information Bottleneck. In *Proceedings of Neural Information Processing Systems NIPS,* 2001.

[94] Slonim N. and Tishby N., Document Clustering Using Word Clusters via the Information Bottleneck Method. In proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 208-215, 2000.

[95] Shin K., Han S. and Gelbukh A. Advanced clustering technique for medical data using semantic information, 322-331, 2004.

[96] Shvaiko P. and Euzenat J. A survey of schema-based matching approaches. Journal on Data Semantics IV, pp 146-171, 2005.

[97] Steinbach M., Karypis G., Kumar V. A comparison of document clustering techniques. In *proceedings of KDD workshop on text mining.* 2000.

[98] Steyvers M., Smyth P., Rosen-Zvi M, Griffiths T., Probabilistic author-topic models for information discovery. In proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, pp 306-315, 2004.

[99] Steinbach M. and Karypis G. and Kumar V., A comparison of document clustering techniques. In *proceedings of KDD Workshop on Text Mining,* 2000.

[100] Strehl A. and Ghosh J., Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions, In *Proceedings of Machine Learning Journal,* vol. 3, pp. 583-618, 2002.

[101] Strube M. and Ponzetto S.P. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In Proceedings of AAAI, 2006.

[102] Tan P.N., Steinbach M., Kumar V., Introduction to Data Mining, *Addison-Wesley Longman Publishing Co,* 2005.

[103] TechTC - Technion Repository of Text Categorization Datasets - http://techtc.cs.technion.ac.il/

[104] Termier A., Rousset MC, Sebag M, Combining statistics and semantics for word and document clustering, In *proceedings of IJCAI,* 2001, 49-54.

[105] Tibshirani R., Walther G., and Hastie T., Estimating the Number of Clusters in a Data Set via Gap Statistic, In *Journal of Royal Statistical Soc. B,* vol. 63, pp. 411-423, 2001.

[106] Tishby Naftali, Periera F. Bialek W., The Information Bottleneck Method. In Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing. 368-377, 1999.

[107] Topchy A., Jain A.K., Punch W., A mixture model for clustering ensembles, In *Proceedings of SIAM Conference on Data Mining,* pp 379–390, 2004.

[108] Topchy A., Jain A.K., and Punch W., Combining Multiple Weak Clusterings, In *Proceedings of IEEE International Conference on Data Mining,* pp. 331-338, 2003.

[109] Wang P. and Domeniconi C. Building Semantic Kernals for text classification using Wikipedia. In proc. Of KDD, pp 24-27, 2008.

[110] Wang P., Hu J., Zeng H. J., Chen L, Chen Z. Improving text classification by using encyclopedia knowledge. In Proceedings of ICDM 2007, 332-341.

[111] Wagstaff A., Cardie C., Rogers S., and Schroedl S., Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning* 2001, 577–584.

[112] Wang Y., Hodges J., Document Clustering with Semantic Analysis. In Proc. of 39th Annual Hawaii International Conference. 54c-54c, 2006.

[113] A. Weingessel, E. Dimitriadou, and K. Hornik, An Ensemble Method for Clustering, In Proceedings of the 3rd International Workshop on Distributed Statistical Computing, 2003.

[114] Wu Z. and Palmer M. Verb Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 133-138, 1994.

[115] Yoo I., Hu X., Song I., Integration of Semantic-based Bipartite Graph Representation and Mutual Refinement Strategy for Biomedical Literature Clustering. . In proc. of 12th ACM KDD, pp 791-796, 2006.

[116] Zhang Zhao, and Ye N., Locality preserving multimodal discriminative learning for supervised feature selection. In Knowledge and Information Systems KAIS Journal, 2010.

[117] Zhang X., Jing L., Hu x.,Ng M., Zhou X. A comparative study of Ontology Based Term Similarity Measures on Pubmed Document Clustering. In proc. Of 12th International Conference on Database Systems for Advanced Applications DASFFA, 2007

[118] Zhang X., Zhou X. and Hu X., Semantic Smoothing for Model-based Document Clustering. In Proceedings of International Conference on Data Mining ICDM, pp 18-22, 2006.

[119] Zhao Y. and Karypis G. Criterion functions for document clustering: Experiments and analysis. *Technical Report TR* 01-40. 2001.

[120] Zhong, S. and Ghosh J., Generative model-based document clustering: a comparative study. In Proceedings of Knowledge and Information Systems, pp 374-384, 2005

[121] Zhou X., Zhang X. and Hu X. Semantic smoothing of Document Models for Agglomerative Clustering. In Proceedings of the 20th International Joint Conference on Artificial Intelligence IJCAI, 2007.

[122] Zhu S., Zeng J, Mamitsuka H., Enhancing MEDLINE clustering by incorporating MeSH semantic similarity, Bioinformatics, 2009.

MICHIGAN STATE UNIVERSITY LIBRARIES
3 1293 03063 8518