Exploiting Wikipedia as External Knowledge for Document Clustering

Xiaohua Hu[1], Xiaodan Zhang[1], Caimei Lu[1], E. K. Park[2], Xiaohua Zhou[1]
[1]College of Information Science and Technology, Drexel University, Philadelphia, PA 19104, USA
[2]School of Computing and Engineering, University of Missouri at Kansas City, Kansas City, MO 64110, USA
{xiaohua.hu, xiaodan.zhang, caimei.lu, xiaohua.zhou}@drexel.edu, ekpark@umkc.edu

ABSTRACT
In traditional text clustering methods, documents are represented as "bags of words" without considering the semantic information of each document. For instance, if two documents use different collections of core words to represent the same topic, they may be falsely assigned to different clusters due to the lack of shared core words, although the core words they use are probably synonyms or semantically associated in other forms. The most common way to solve this problem is to enrich document representation with the background knowledge in an ontology. There are two major issues for this approach: (1) the coverage of the ontology is limited, even for WordNet or Mesh, (2) using ontology terms as replacement or additional features may cause information loss, or introduce noise. In this paper, we present a novel text clustering method to address these two issues by enriching document representation with Wikipedia concept and category information. We develop two approaches, *exact match* and *relatedness-match*, to map text documents to Wikipedia concepts, and further to Wikipedia categories. Then the text documents are clustered based on a similarity metric which combines document content information, concept information as well as category information. The experimental results using the proposed clustering framework on three datasets (20-newsgroup, TDT2, and LA Times) show that clustering performance improves significantly by enriching document representation with Wikipedia concepts and categories.

Categories and Subject Descriptors
I.5.3 [**Pattern recognition**]: Clustering – *algorithms, similarity measures.*

General Terms
Algorithms, Experimentation

Keywords
Text Clustering, Wikipedia, Document Representation

KDD'09, June 28-July 1, 2009, Paris, France.

1. INTRODUCTION
Traditional clustering algorithms are usually based on the BOW (Bag of Words) approach. A notorious disadvantage of the BOW model is that it ignores the semantic relationship among words. As a result, if two documents use different collections of core words to represent the same topic, they can be assigned to different clusters, although the core words they use are probably synonyms or semantically associated in other forms. One way to resolve this problem is to enrich document representation with the background knowledge represented by an ontology.

An ontology usually includes at least three components: concepts, attributes, and the relationships among concepts. All of them can be used for document representation and clustering. The most common way of applying ontologies for clustering is to match ontology concepts to the topical terms appearing in the documents. Then the matched ontology concepts are either used as replacement or introduced as additional features to the original text. Further, the attributes of and relationships among the ontology terms can be exploited for clustering.

However, a major problem of this approach is that it is usually difficult to find a comprehensive ontology which can cover all the concepts mentioned in a collection, especially when the documents to be clustered are from general domain. Previous research has adopted WordNet [4, 5] and Mesh [12, 13] as the external ontology for text enrichment. However, they all have limited coverage. Another problem is that using ontology terms either as replacement or additional features has its disadvantages. While replacing original content with ontology terms may cause information loss, especially when the coverage of the ontology is limited, adding ontology terms to the original document vector can bring data noise into the dataset. Therefore, in order to enhance text clustering by leveraging ontology semantics, two issues need to be addressed: an ontology which can cover the topical domain of individual document collections as completely as possible; and a proper matching method which can enrich the document representation by fully leveraging ontology terms and relations but without introducing more noise.

This paper aims to address both issues. In terms of ontology, we rely on Wikipedia concepts and categories for document enrichment. Wikipedia has become the largest electronic knowledge repository on the web with millions of articles contributed collaboratively by volunteers. Unlike other standard ontologies, such as WordNet and Mesh, Wikipedia itself is not a structured thesaurus. However, it is much more comprehensive

and up to date. Moreover, it is well-formed. In Wikipedia, each article only describes a single topic. The title of each article is a succinct phrase that resembles an ontology term. Equivalent concepts are grouped together by redirected links. Meanwhile, it contains a hierarchical categorization system, in which each article belongs to at least one category. All these features make Wikipedia a potential ontology which can be exploited for enriching text representation and enhancing text clustering.

As for how to integrate ontology concepts into the document representation and clustering process, in this paper, we propose two approaches for mapping ontology concepts to the documents. The first approach, called exact-match, is a dictionary-based approach. It maps the topical terms present in the documents directly to Wikipedia concepts. It is especially useful when Wikipedia concepts can cover most of the topic terms in a collection. The second mapping approach is called relatedness-match. Instead of mapping Wikipedia concepts to each document directly, this approach builds the connection between Wikipedia concepts and each document based on the contents of Wikipedia articles. This approach is more useful when Wikipedia concepts cannot fully cover the topical domain of a collection. After the mapping process, each document is associated with a set of concepts. Then based on the hierarchical structure of Wikipedia, each document is further mapped to a set of Wikipedia categories. Finally, the text documents are clustered based on a similarity metric which combines document content information, concept information as well as category information.

The proposed Wikipedia-based clustering framework is evaluated on three datasets: 20-newsgroups, TDT2, and LA Times. We use both agglomerative and partitional clustering for experiments and the traditional BOW model as the baseline. The results show that, in agglomerative clustering method, enriching document representation with Wikipedia concepts and categories by both exact-match and relatedness match can significantly improve the clustering performance. However, the results of partitional clustering vary among different datasets and depend on the matching scheme adopted.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 introduces the proposed method of utilizing Wikipedia concepts and categories to improve document clustering. In Section 4, we present and discuss experimental results. Finally, we conclude the paper in section 5.

2. RELATED WORKS

Recently, there is a growing amount of research on how to utilizing Wikipedia to enhance text mining tasks, such as information retrieval [7, 8], text classification [2, 11] and clustering [1, 6].

Milne et al. [8] develop a search engine that works on the basis of the thesauri derived from Wikipedia. The hyperlinks, redirect links and hierarchical relations within Wikipedia are exploited to build the thesauri that are specific to individual collections. Based on the thesauri, the search engine can expand queries automatically and guide users to improve their queries during the search process. Li et al. [7] use Wikipedia category information to improve weak ad-hoc queries. After the initial search with a weak query, the returned articles are re-ranked based on the linear combination of their original ranking score and Wikipedia category score. Then a certain number of terms are selected from top-ranked articles to expand the search query.

Phan et al. [9] present a framework for discovering hidden topics from large-scale data collections to resolve the data sparsity problem in short text classification. Instead of using human category information in Wikipedia, they use Gibbs sampling and LDA to sample topics from both large-scale data collection and a sparse testing dataset. Each testing document is classified based on a vector combining both content and topic information. Although the approach provides a different perspective on using large-scale text collection, it does not fully utilize useful information embedded in Wikipedia such as the category, link information, etc. Moreover, the sampling process can be very time consuming and the sampled topics are time sensitive to Wikipedia snapshots.

Gabrilovich and Markovitch [2, 3] propose a method to improve text classification performance by enriching document representation with Wikipedia concepts. The mapping between each document and Wikipedia concepts is achieved through a feature generator which acts like a retrieval engine. It receives a text fragment, which can be words, sentence, paragraph, or the whole document, and outputs the most relevant Wikipedia articles to the text fragment. The titles of the retrieved Wikipedia articles are further filtered and those with high discriminative capacity are used as additional features to enrich the representation of the corresponding documents. Empirical evaluation shows that their method can greatly improve classification performance. However, the multi-resolution feature generation procedure they apply for mapping Wikipedia concepts requires high processing efforts, because each document needs to be scanned multiple times. And it produces too many Wikipedia concepts for each document. Especially, when the text fragments used for retrieving Wikipedia articles are generic words or sentences, this procedure only introduces noise. Although the authors apply a filtering step to eliminate extraneous features, it further increases the processing efforts and time. In our method of relatedness-match, we only use document words to retrieve Wikipedia articles. However, each document word is weighted based on their tfidf value. Thereby, Wikipedia concepts retrieved by important words with high tfidf value are ranked higher than those retrieved through unimportant words.

Wikipedia has also been applied for text clustering. Banerjee et al. [1] use a method similar to that applied in [2] for clustering short texts. But different from [2], they use query strings created from document texts to retrieve relevant Wikipedia articles. The titles of top-ranked Wikipedia articles serve as additional features for clustering Google news.

The method in both [1] and [2] only augment document representation with Wikipedia concepts without considering the hierarchical relationship embedded in Wikipedia. In our method, we also integrate Wikipedia category information into document representation based on the hierarchical structure of Wikipedia. We believe that integrating high-level category information can further improve clustering performance by introducing more background knowledge into the clustering process.

The Wikipedia category information has also been utilized in [6] and [11] for text clustering and classification respectively. Besides, they also extend the Wikipedia concept vector for each

document with synonyms and associative concepts based on the redirect links and hyperlinks in Wikipedia. Their methods to a great extent leverage the abundant structural information within Wikipedia. However, they all rely on an exact phrase matching strategy for mapping text documents to Wikipedia concepts. This strategy is limited by the terms appearing in the documents and the coverage of Wikipedia concepts or article titles. For instance, if the topical terms used in a document are not exactly the same as any Wikipedia concept but synonymous to some of them, then the Wikipedia concepts which have the same meaning with the topical terms would not be mapped to the documents. In our paper, to solve this problem, we adopt another mapping strategy called relatedness-match, which does not merely using Wikipedia article titles for matching but also considering the content of the whole Wikipedia articles during the matching process.

3. FRAMWORK OF WIKIPEDIA-BASED CLUSTERING

The framework of our method for leveraging Wikipedia concept and category information to improve document clustering is presented in Figure 1.

We first define two concept mapping schemes: exact-match and relatedness-match. Then, based on the two mapping schemes, we construct concept feature vector and category feature vector for each document. The document content vector W_n, concept vector C_n and category vector Cat_n are linearly combined to measure document similarity. Finally, with the new similarity metric, the documents are clustered using agglomerative approach and partitional approach respectively.

3.1 Mapping Documents to Wikipedia Concepts and Categories

The mapping process includes three steps: (1) build the connection between Wikipedia concepts and categories; (2) map each document into a vector of Wikipedia concepts; (3) match each document to a set of Wikipedia categories. Each step generates a matrix (see Figure 2). The concept-category matrix is created intuitively based on the connection between concepts and categories which is explicit in Wikipedia. The document-concept matrix is built through two matching schemes: exact-match and relatedness-match. Finally, the document-category matrix is created on the basis of concept-category matrix and document-concept matrix.

3.2 Concept Mapping Schemes

A proper matching method is crucial for ontology-based text clustering. In our research, we adopt two different match schemes (exact-match and relatedness-match) for mapping documents to Wikipedia concepts. The details of each mapping scheme are described below.

3.2.1 Exact-Match Scheme

By exact-match scheme, each document is scanned to find Wikipedia concepts, which are mostly short phrases. The searched Wikipedia concepts are used to comprise the concept vector of the corresponding document. An issue of exact-match is how to map synonymous phrases to the same concept. We address

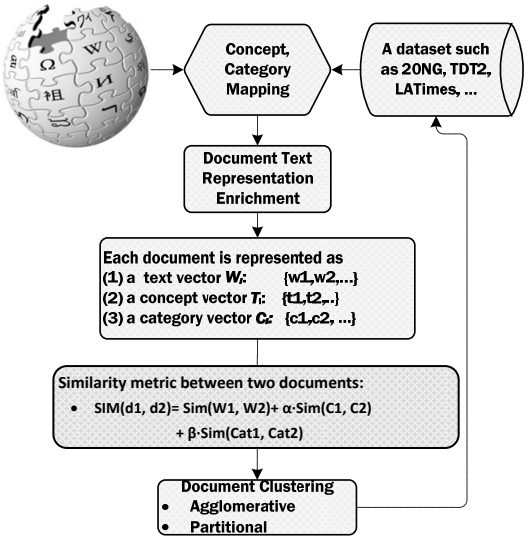

Figure 1. The framework of leveraging Wikipedia for document clustering

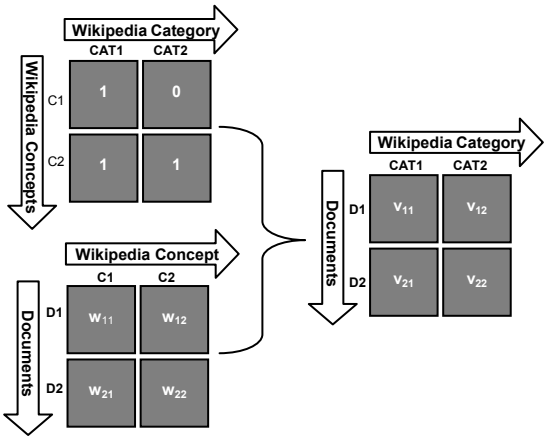

Figure 2. Mapping documents to Wikipedia concepts and category

this problem by using the redirect links in Wikipedia. In Wikipedia, each topic is described by only one article. A preferred phrase is chosen as the title of the article. All other phrases representing the same topic are redirected to the same article. Based on this feature, we construct a dictionary. Each entry in the dictionary corresponds to a topic covered by Wikipedia. Each entry includes not only the preferred Wikipedia concept which is used as the title of the article, but also all redirected concepts representing the same topic. Based on this dictionary, both preferred concepts and redirected concepts are retrieved from documents. However, only preferred concepts are used to build the concept vector for each document. The weight of each preferred concepts equals to the frequency of itself plus the frequencies of all the concepts redirected to it. In this way, we get a document-concept matrix, whose values are the frequencies of each concept appearing in a document. Based on this frequency matrix, we further calculate the document-concept TFIDF matrix, which is used to measure the similarity between two documents'

concept vectors. Compared to other matching technique, exact-match is very efficient. However, it always has low recall. It produces good result only when Wikipedia has a good coverage of the phrases appearing in a dataset.

3.2.2 Relatedness-Match Scheme

By exact-match, only the concepts which explicitly appear in a document are extracted and used to construct the concept vector of the document. In other words, when the topical terms used in a document can not exactly match the Wikipedia concepts denoting the same topic, they cannot be extracted. In order to resolve this problem, we adopt another match scheme called relatedness-match.

Relatedness-match consists of two steps. First, we create a Wikipedia term-concept matrix (See Figure 3) from Wikipedia article collection. Thus each word token is represented by a concept vector. The values of the vector are TFIDF scores, which denote the relatedness between the term and each Wikipedia concept. A word may appear in a huge number of Wikipedia articles. In order to discard insignificant concepts and improve processing efficiency, for each word, we only choose top *k* concepts with highest *TFIDF* scores. In this study, we set *k* to 5.

Table 1. Wikipedia term-concept (article) matrix

		Wiki concepts C			
		c_1	c_2	…	c_N
Wiki article terms *W*	w_1	val_{11}	val_{12}	…	val_{1N}
	w_2	val_{21}	val_{22}	…	val_{2N}
	…	…	…	…	…
	w_N	val_{N1}	val_{N2}	…	val_{NN}

Second, the word-concept matrix is used as a bridge to associate documents with Wikipedia concepts. The relatedness of a Wikipedia concept to a given document is calculated using equation (1).

$$r_k^{d_j} = \sum_{w_i \in d_j} tfidf_{d_j}^{w_i} \bullet tfidf_{c_k}^{w_i} \quad (1)$$

where $d_j \in D$ (a document collection) and $c_k \in C$ (all Wikipedia preferred concepts). The procedure of calculating the relatedness of concept c_k to document d_j is as follows. For each word such as w_i in document d_j, we calculate its *TFIDF* scores in both d_j and c_k. The two scores specify the importance of word w_i to document d_j and concept c_k, respectively. Then we use the product of two *TFIDF* values as the relatedness score of concept c_k to document d_j through word w_i. By summing up the relatedness score of concept c_k to document d_j through each word in document d_j, we get the final relatedness score $r_k^{d_j}$ of concept c_k to document d_j. Then, we select top *M* concepts with highest relatedness score for each document. In this experiment, M is set to 200. Finally, the concept relatedness score vector for each document is normalized.

Compared to exact-match, this method is more time consuming, it help identify relevant Wikipedia concepts which are not explicitly present in a document. It is especially useful when Wikipedia concepts have less coverage for a dataset.

3.3 Category Mapping

After concept mapping, a document-concept matrix is generated for each collection. Based on the document-concept matrix and the hierarchical relation between Wikipedia concept and category, we derive the document-category matrix (see Figure 2).

If the document-concept matrix is created through exact-match, a document-category frequency matrix is first derived from the document-concept frequency matrix by replacing each concept with its corresponding categories. The frequency of a category is the frequency of the concept belonging to it. If a category is mapped to a document through more than one concept, the sum of the frequencies of these concepts is the category's frequency. Based on the generated document-category frequency matrix, we further derive the document-category TFIDF matrix, which is used to measure the similarity between any two documents' category vectors.

If the document-concept matrix is developed through relatedness-match, to get the document-category matrix, we replace each concept with its corresponding categories and all these categories share the same normalized relatedness score as the concept. If a category is mapped to a document through more than one concept, its relatedness score to the document is the sum of the scores of all these concepts. The derived document-category matrix denotes the relatedness of each category to each document.

3.4 Document Clustering

3.4.1 Agglomerative Clustering

Agglomerative clustering approaches initially consider each document as a cluster and repeatedly merge pairs of clusters with shortest distance until only one cluster is formed covering all the documents. The distance measure between two clusters can be implemented in many ways including *single linkage*, *complete linkage*, and *average linkage*. In our experiment, when using standard vector cosine similarity as document similarity measure, both single linkage and average linkage suffer a severe chaining problem on all three testing datasets. Therefore, we use complete linkage as cluster distance measure. With complete linkage criterion, the distance of two clusters is defined as the maximum distance between one document in the first cluster and the other in the second cluster. In our method, besides word vector, a document is also represented by concept vector or category vector, or both of them. When calculating similarity between two documents, we combine the similarity values calculated using these vectors (see equation (2)).

$$sim(d_m, d_n) = sim(d_m, d_n)^{word} + \alpha \bullet sim(d_m, d_n)^{concept} + \beta \bullet sim(d_m, d_n)^{category} \quad (2)$$

, where coefficient α and β indicates the importance of concept vector and category vector in measuring the similarity between two documents.

3.4.2 Partitional Clustering

Partitional clustering approaches iteratively calculate the cluster centroids and reassign each document to the closest cluster until no document can be reassigned. Spherical k-means is one of these algorithms and most widely used for text clustering. Therefore, we apply spherical k-means for partitional approach. In our method, the distance from a document to a cluster centroid is calculated based on the content similarity as well as concept similarity or category similarity, or both of them.

$$sim(d_m, centroid_k) = sim(d_m, centroid_k)^{word} + \alpha \bullet sim(d_m, centroid_k)^{concept} + \beta \bullet sim(d_m, centroid_k)^{category} \quad (3)$$

, where α and β quantifies the influence of the concept and category information on document clustering.

Since the clustering result of k-means is influenced by the initial selection of cluster centroids. For each evaluation based on K-means, we run ten times with random initialization and take the average as the final clustering result. In comparative experiment, each run has the same initialization.

4. EXPERIMENTS

4.1 Wikipedia Data

Wikipedia release its database dumps periodically, which can be downloaded from http://download.wikipedia.org. The Wikepedia dump we use contains 911,028 articles and about 29000 categories after pre-processing and filtering.

4.2 Clustering Dataset

We perform clustering experiments on three datasets: TDT2, LA Times (from TREC), and 20-newsgroups (20NG). We selected 7,094 documents in TDT2 that have a unique class label, 18,547 documents from top ten sections of LA Times, and all 19,997 documents in 20-newsgroups. The ten classes selected from TDT2 are *20001, 20015, 20002, 20013, 20070, 20044, 20076, 20071, 20012, and 20023*. The ten sections selected from LA Times are *Entertainment, Financial, Foreign, Late Final, Letters, Metro, National, Sports, Calendar, and View*. All 20 classes of 20NG are used for testing.

For efficiency, we adopt a special evaluation approach. For each dataset, we create five small datasets. Each small dataset is created by randomly picking 100 documents from each selected class of a given dataset and then merge them into a big pool. The five small datasets are clustered separately, and the average of their results is viewed as the clustering result for the whole dataset.

4.3 Evaluation Metrics

Cluster quality is evaluated by three metrics, *purity* [14], *F-score* [10], and *normalized mutual information (NMI)* [15]. Purity assumes that all samples of a cluster are predicted to be members of the actual dominant class for that cluster. F-score combines the information of precision and recall which is extensively applied in information retrieval. NMI is an increasingly popular measure of clustering quality. It is defined as the mutual information between the cluster assignments and a pre-existing labeling of the dataset normalized by the arithmetic mean of the maximum possible entropies of the empirical marginals, i.e.

$$NMI(X,Y) = \frac{I(X;Y)}{(\log k + \log c)/2} \quad (4)$$

, where X is a random variable for cluster assignments, Y is a random variable for the pre-existing labels on the same data, k is the number of clusters, and c is the number of pre-existing classes. A merit of NMI is that it does not necessarily increase when the number of clusters increases. All the three metrics range from 0 to 1, and the higher their value, the better the clustering quality is.

4.4 Clustering Schemes under Comparison

In both agglomerative and partitional clustering approaches, we use the clustering approach based on word-only vectors as the baseline. Other approaches based on different linear combinations of word vector, concept vector and category vector are listed in Table 2.

Table 2. Clustering schemes based on different combinations of vectors

Notation	Explanation
Word	Clustering solely based on word vector
Concept	Clustering solely based on concept vector
Category	Clustering solely based on category vector
Word_Concept	Clustering based on the linear combination of word vector and concept vector
Word_Category	Clustering based on the linear combination of word vector and category vector
Concept_Category	Clustering based on the linear combination of concept vector and category vector
Word_Concept_Category	Clustering based on the linear combination of word vector, concept vector and category vector

The parameter α and β in equation (2) and (3) are set in the following way:

- For clustering based on Word_Concept scheme, β is set to zero and α is set to 0.1, 0.2, ···, 1.0 respectively. We take the average result of the ten runs as the final clustering results for Word_Concept scheme.
- For clustering based on Word_Category scheme, α is set to zero and β is set to 0.1, 0.2, ···, 1.0 respectively. The average result of the ten runs is used as the final clustering results for Word_Category scheme.
- For Word_Concept_Category scheme, α is set to the value which produces best results for Word_Concept based clustering, and β is set to the value that generates best results for Word_Category based clustering.

4.5 Agglomerative Clustering Results

Table 3 shows the results of agglomerative clustering using two different match schemes: Exact-match (EM) and relatedness-match (RM).

The bold values in table 3 are improved results compared to the baseline. The "*" indicates the improvement is significant according to the paired-sample T-test at the level of $p<0.05$. These symbols are applied in all following experimental result tables.

Table 3. Agglomerative clustering results on three datasets

	20 News Group					
	NMI		**F-Score**		**Purity**	
Match Scheme	**EM**	**RM**	**EM**	**RM**	**EM**	**RM**
Word (BaseLine)	0.144		0.146		0.132	
Concept	0.134	**0.157**	0.102	0.062	0.103	0.081
Cateogory	0.111	**0.160***	0.128	0.111	**0.143**	0.114
Concept_Category	0.131	**0.148**	0.146	0.084	**0.160**	0.095
Word_Concept	0.144	**0.150**	**0.153**	**0.168**	**0.136**	**0.148**
Word_Category	**0.166***	**0.171***	**0.189***	**0.209***	**0.201***	**0.180***
Word_Concept_Category	**0.166***	**0.154**	**0.196***	**0.195***	**0.206***	**0.165**
	LATimes					
	NMI		**F-Score**		**Purity**	
Match Scheme	**EM**	**RM**	**EM**	**RM**	**EM**	**RM**
Word (BaseLine)	0.048		0.066		0.124	
Concept	**0.060**	**0.073**	0.057	0.044	0.120	0.113
Cateogory	**0.071***	**0.053**	**0.174**	0.054	0.111	0.118
Concept_Category	**0.073**	**0.054**	**0.177**	0.054	**0.202**	0.118
Word_Concept	**0.051**	**0.052**	0.064	**0.072**	0.124	**0.128**
Word_Category	**0.101***	**0.049**	**0.210***	**0.097***	**0.238***	**0.142***
Word_Concept_Category	**0.103***	**0.052**	**0.204***	**0.100***	**0.232***	**0.144***
	TDT2					
	NMI		**F-Score**		**Purity**	
Match Scheme	**EM**	**RM**	**EM**	**RM**	**EM**	**RM**
Word (BaseLine)	0.537		0.622		0.600	
Concept	0.296	0.372	0.398	0.483	0.368	0.463
Cateogory	**0.577***	0.448	**0.637**	0.539	**0.649***	0.549
Concept_Category	**0.581***	0.444	**0.656**	0.543	**0.659***	0.560
Word_Concept	**0.563**	**0.609**	**0.637**	**0.689***	**0.620**	**0.678***
Word_Category	**0.695***	**0.660***	**0.754***	**0.721***	**0.769***	**0.737***
Word_Concept_Category	**0.675***	**0.661***	**0.734***	**0.726***	**0.751***	**0.747***

From Table 3, we can see that the scheme Word_Category and Word_Concept_Category always get the best results across all three datasets. In most cases, they can significantly improve the performance of clustering. However, out of our expectation, Word_Concept_Category does not perform better than Word_Category. Moreover, although in most cases Word_Concept scheme can also improve clustering results, the improvement are not significant. Sometimes it even performs worse than the baseline. This indicates that integrating Wikipedia concept information into clustering process does not necessarily improve clustering performance. This conclusion can be further confirmed by examining the clustering results using Concept vector alone. In most cases, clustering only based on concept information performs worse than the baseline. On the other hand, Wikipedia category information is much more valuable for improve clustering performance. In general, combining word vector and Wikipedia category vector can significantly improve clustering results. For instance, according to NMI, for 20 Newsgroup, Word_Category achieves 15.3% and 18.8% increase in performance with exact-match and relatedness-match respectively; for TDT2, Word_Category improves the performance by 29.4% and 22.9% with exact-match and relatedness-match respectively. Besides, clustering solely based on category vector most times performs better than clustering solely using concept vector, and have better or close performance to the baseline. This observation is especially true for the dataset TDT2. We also tested clustering based on category and cluster vector together (Concept_Category). For LATimes and TDT2, it performs better than using either category information or concept information alone. However, for 20 Newsgroup, its performance is quite unstable.

In summary, our experimental results of agglomerative clustering show that category information is more useful than concept information for improving clustering results. We think the reason is that the Wikipedia concept collection we applied for experiment still contains too much noise. By integrating concept information into document presentation, we also introduce noise to the clustering process. Another reason is that we do not disambiguate concept senses during the concept mapping process. This may further decrease the discriminative capacity of the concept vectors created for the documents. Compared to concept, category information is much less suffered from noise, and more accurate and informative.

It is not apparent which match scheme is better. Their effect on clustering results always depends on the datasets and clustering schemes. For instance, according to NMI, for 20 Newsgroup, relatedness-match based clustering outperforms exact-match based clustering across all schemes except Word_Concept_Category. For the other two datasets, exact-match performs better than relatedness-match in most cases.

4.6 Partitional Clustering Results

Table 4 lists the results of partitional clustering based on different vector schemes and using two different match schemes. We can see that the effect of category information and cluster information on clustering results is not as significant as in agglomerative clustering. We think this is because, in K-means, category vector and concept vector are not used to measure the similarity between two documents, but used to calculate the distance between a document and a cluster centroid. Accordingly, category information and concept information are not utilized in full scale. Even so, we can still see the contribution of category information to clustering results. For 20 Newsgroup, Word_Category scheme still significantly improves the clustering result. The F-Score and Purity of Word_Concept_Category based clustering are also significantly improved. For TDT2, Word_Concept_Category produces the best clustering results.

It is also notable that for dataset 20 Newsgroup, relatedness-match always produces better results than exact-match. But for the other two datasets, LATimes and TDT2, exact-match always outperforms relatedness-match.

Table 4. Partitional clustering results on three datasets

	20 News Group					
	NMI		F-Score		Purity	
Match Scheme	EM	RM	EM	RM	EM	RM
Word (BaseLine)	0.390		0.382		0.411	
Concept	0.288	0.313	0.312	0.372	0.302	0.401
Cateogory	0.291	0.326	0.332	**0.383**	0.341	0.391
Concept_Category	0.287	0.322	0.333	0.354	0.317	0.388
Word_Concept	0.390	0.383	0.380	0.382	0.411	0.411
Word_Category	**0.409***	**0.429***	**0.402***	**0.412***	**0.430***	**0.442***
Word_Concept_Category	**0.398**	**0.412**	**0.400***	**0.418***	**0.429***	**0.442***
	LATimes					
	NMI		F-Score		Purity	
Match Scheme	EM	RM	EM	RM	EM	RM
Word (BaseLine)	0.188		0.317		0.328	
Concept	0.186	0.082	0.312	0.253	**0.333**	0.251
Cateogory	0.185	0.097	0.315	0.241	0.327	0.249
Concept_Category	**0.190**	0.112	0.310	0.242	**0.329**	0.245
Word_Concept	0.159	0.128	0.292	0.264	0.304	0.275
Word_Category	**0.194**	0.179	**0.325**	0.312	**0.335**	0.322
Word_Concept_Category	**0.189**	0.140	**0.319**	0.276	**0.330**	0.286
	TDT2					
	NMI		F-Score		Purity	
Match Scheme	EM	RM	EM	RM	EM	RM
Word (BaseLine)	**0.790**		0.825		0.848	
Concept	0.556	0.447	0.622	0.522	0.647	0.544
Cateogory	0.577	0.448	0.637	0.539	0.649	0.549
Concept_Category	0.543	0.442	0.630	0.523	0.643	0.545
Word_Concept	0.787	0.766	0.815	0.792	0.840	0.819
Word_Category	**0.804**	0.737	**0.830**	0.720	**0.854**	0.763
Word_Concept_Category	**0.802**	**0.804**	**0.833**	**0.846**	**0.854**	**0.876**

5. CONCLUSION AND FUTURE WORK

In this paper, we present a general framework for leveraging Wikipedia concept and category information to improve text clustering performance. Based on two different mapping techniques, exact-match and relatedness-match, we are able to create a Wikipedia concept vector and a Wikipedia category vector for each document in a collection. The concept vector and category vector provide background knowledge about a document. They are linearly combined with text word vector to measure document similarity.

The propose framework is tested with two clustering approaches (agglomerative and partitional clustering) on three datasets: 20NG, LATimes and TDT2. In order to comprehensively evaluate the effect of Wikipedia concept and category information on clustering performance, we experiment seven different clustering schemes—Concept, Category, Word_Concept, Word_Category, Concept_Category, and Word_Concept_Category. Based on the empirical results, we can draw the following conclusions: (1) Category information is most useful for improving clustering results. In both agglomerative clustering and partitional clustering, combining category information with document content information generates the best results in most cases. Compared to the baseline scheme, it can significantly improve clustering performance for all three datasets when using agglomerative clustering approach and for dataset 20 NewsGoup when using partitional clustering. (2) Clustering based on all three document vectors (word vector, concept vector, category vector) also gets significantly better results than the baseline. However, it does not outperform clustering based only on word vector and category vector. (3) Concept information is not as useful as category information for improving clustering performance due to the noise information it contains and sense ambiguity problem. (4)The effect of category and concept information on k-means clustering is not as significant as it on agglomerative clustering. But, in most cases, Word_Category based clustering still achieves best performance among all clustering schemes. (5) The effect of the two mapping schemes depends on the dataset, quality metric and clustering approach. Based on the results of partitional clustering, exact-match is more effective than relatedness-match for dataset LATimes and TDT2, but on the contrary for 20 Newsgroup.

We believe that our findings can be extended to other applications based on document similarity measurement, such as information retrieval and text classification. For future work, we will further improve our concept mapping techniques, such as introducing sense disambiguation functions into the concept mapping process. Moreover, we will explore how to utilize the link structure among Wikipedia concepts for document clustering.

6. ACKNOWLEDGEMENTS

This work is supported in part by NSF Career Grant IIS 0448023, NSF CCF 0514679, PA Dept of Health Tobacco Settlement Formula Grant (No. 240205 and No. 240196), and PA Dept of Health Grant (No. 239667).

7. REFERENCES

[1] Banerjee, S., Ramanathan, K. and Gupta, A. 2007. Clustering short texts using Wikipedia. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (Amsterdam, The Netherlands, July 23-27, 2007). ACM Press, New York, NY, 787-788.

[2] Gabrilovich, E. and Markovitch, S. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In Proceedings of the 21st National Conference on Artificial Intelligence. (Boston, MA, July 16–20, 2006). 1301-1306.

[3] Gabrilovich, E. and Markovitch, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence. (Hyderabad, India, January 6-12, 2007). 1606-1611.

[4] Hotho, A., Staab, S.and Stumme, G. 2003. Wordnet improves text document clustering. In Proceedings of Semantic Web Workshop, the 26th annual International ACM SIGIR Conference. (Toronto, Canada, Jul. 28-Aug.1, 2003).

[5] Hotho, A., Maedche, A. and Staab, S. Text Clustering Based on Good Aggregations, In Proceedings of the 2001 IEEE International Conference on Data Mining. (San Jose, CA, Nov. 29-Dec.02, 2001,). IEEE Computer Society, Washington, DC, 607-608.

[6] Hu, J., Fang, L., Cao, Y., et al. Enhancing Text Clustering by Leveraging Wikipedia Semantics. In Proceedings of the 31st annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (Singapore, July 20 – 24, 2008). ACM Press, New York, NY, 179-186.

[7] Li, Y., Luk, W.P.R, Ho, K.S.E., and Chung, R.L.K. 2007. Improving Weak Ad-Hoc Queries using Wikipedia as External Corpus. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (Amsterdam, The Netherlands, July 23-27, 2007). ACM Press, New York, NY, 797 - 798.

[8] Milne, D. 2007. Computing Semantic Relatedness using Wikipedia Link Structure. In Proceedings of the 5th New Zealand Computer Science Research Student Conference. (Hamilton, New Zealand, April 10-13, 2007).

[9] Phan, X., Nguyen, L. and Horiguchi, S. 2008. Learning to Classify Short and Sparse Test & Web with Hidden Topics from large-scale Data collection. In Proceedings of 17th International World Wide Web Conference. (Beijing, China, April 21-25, 2008). ACM Press, New York, NY, 91-100.

[10] Steinbach, M., Karypis, G. and Kumar, V. 2000. A Comparison of document clustering techniques. Technical Report. Department of Computer Science and Engineering, University of Minnesota.

[11] Wang, P. and Domeniconi, C. 2008. Building Semantic Kernels for text classification using Wikipedia. In Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (Nevada, Las Vegas, August 24 – 27, 2008). ACM Press, New York, NY, 713-721.

[12] Yoo, I., Hu, X. and Song, I.-Y. 2006. Integration of semantic-based bipartite graph representation and mutual refinement strategy for biomedical literature clustering. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. (Philadelphia, PA, August 20 – 23, 2006). ACM Press, New York, NY, 791 – 796.

[13] Zhang, X., Jing, L., Hu, X., et al. A Comparative Study of Ontology Based Term Similarity Measures on Document Clustering. In Proceedings of 12th International conference on Database Systems for Advanced Applications. (Bangkok, Thailand, April 9-12, 2007).115-126.

[14] Zhao, Y. and Karypis, G. 2001. Criterion functions for document clustering: experiments and analysis, Technical Report. Department of Computer Science, University of Minnesota.

[15] Zhong, S. and Ghosh, J. 2005. Generative model-based document clustering: a comparative study. Knowledge and Information Systems. 8, 3 (Sep. 2005). 374-384.