Enhancing Text Clustering by Leveraging Wikipedia Semantics

Jian Hu[1], Lujun Fang[2], Yang Cao[3], Hua-Jun Zeng[1], Hua Li[1], Qiang Yang[4], Zheng Chen[1]

[1]Microsoft Research Asia
49 Zhichun Road, Beijing 100080, P.R. China
{jianh, hjzeng, huli, zhengc}@microsoft.com

[2]Fudan University
220 Handan Road,Shanghai 200433, P.R. China
fanglujun@fudan.edu.cn

[3]Shanghai Jiao Tong University
1954 Huashan Road, Shanghai 200030, P.R. China
cybersun@163.com

[4]Hong Kong University of Science & Technology
Clearwater Bay, Hong Kong
qyang@cse.ust.hk

ABSTRACT

Most traditional text clustering methods are based on "bag of words" (*BOW*) representation based on frequency statistics in a set of documents. *BOW*, however, ignores the important information on the semantic relationships between *key* terms. To overcome this problem, several methods have been proposed to enrich text representation with external resource in the past, such as WordNet. However, many of these approaches suffer from some limitations: 1) WordNet has limited coverage and has a lack of effective word-sense disambiguation ability; 2) Most of the text representation enrichment strategies, which append or replace document terms with their hypernym and synonym, are overly simple. In this paper, to overcome these deficiencies, we first propose a way to build a *concept thesaurus* based on the semantic relations (synonym, hypernym, and associative relation) extracted from Wikipedia. Then, we develop a unified framework to leverage these semantic relations in order to enhance traditional content similarity measure for text clustering. The experimental results on Reuters and OHSUMED datasets show that with the help of Wikipedia thesaurus, the clustering performance of our method is improved as compared to previous methods. In addition, with the optimized weights for hypernym, synonym, and associative concepts that are tuned with the help of a few labeled data users provided, the clustering performance can be further improved.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology-*Clustering design and evaluation*.

General Terms

Algorithms, Experimentation, Performance, Human Factors

Keywords

Text Clustering, Wikipedia, Thesaurus, Similarity Measure

1. INTRODUCTION

The exponential growth of online document in World Wide Web has raised an urgent demand for efficient, high-quality text clustering algorithms for fast navigation and browsing of users based on better document organizations. However, the traditional document clustering algorithms have been based on a variation of the "bag of words" (*BOW*) approach, which represents the documents with features as weighted occurrence frequencies of individual terms.

The *BOW* representation is limited as it only counts in the term frequency statistics in the documents and ignores the important information of the semantic relationships between *key* terms. Thus, the distance measure of text clustering based on "*BOW*" cannot reflect the actual distance of two documents. As the clustering performance is heavily relied on the distance measure of document pairs, finding an accurate distance measure which can break the limitation of "*BOW*" is crucial for improving text clustering performance. Several works have been done to exploit external resource to enrich text document representation. [1][16][17][24] utilize term ontology structured from WordNet [19] to improve the *BOW* text representation. Among them, Hotho *et al.* [1] adopts various strategies to enrich text documents representation with synonym and hypernym from WordNet, and experimental results showed some improvements in clustering performance. Other research works explored the usage of world knowledge bases in the Web such as Open Directory Project (ODP) [20] and Wikipedia to enrich text document representation. Gabrilovich *et al.* [2][3] try to apply feature generation techniques on ODP and Wikipedia to create new features that augment the bag of words. Their application on text classification confirmed that background-knowledge-based features generated from ODP or Wikipedia can help text categorization and Wikipedia is less noise than ODP when used as knowledge base.

However, these approaches have a number of limitations: First, WordNet has limited coverage – WordNet focuses on the usages of common words which are rarely to be the representative words of a document, and is lack of an effective word sense disambiguation method - the description for different senses of a word is quite short. Second, ODP and Wikipedia themselves are not structured thesaurus as WordNet. While enriching documents with features generated by ODP or Wikipedia, they are not as easy as WordNet to handle the problems of synonymy and polysemy, which are two fundamental problems in text clustering. Meanwhile, The structural relations in Wikipedia is not fully used in Gabrilovich *et al.* [2][3]. Finally, most of text representation enrichment strategies of these approaches, which append or replace document terms with their

SIGIR'08, July 20–24, 2008, Singapore.

hypernym and synonym are overly simple - hypernym and synonym should have different importance as compared to the original document content when computing similarity of document pairs in different datasets.

In this paper, we show that by fully leveraging the structural relationship information in Wikipedia, we can enhance the clustering result by obtaining a more accurate distance measure. In particular, we first build an informative and easy-to-use thesaurus from Wikipedia, which explicitly derives the concept relationships based on the structural knowledge in Wikipedia, including synonymy, polysemy, hypernymy and associative relation. The generated thesaurus serves as a control vocabulary that bridges the variety of idiolects and terminologies present in the document corpus. The thesaurus facilitates the integration of the rich encyclopedia knowledge of Wikipedia into text documents, because it resolves synonyms and introduces more general and associative concepts which may help identify related topics between text documents. Also, the coverage of the thesaurus is much larger than manually constructed thesaurus like WordNet, and it provides a richer context for polysemy concept sense disambiguation. We then propose a novel framework to leverage the hierarchical, synonymy and associative semantic relations from a Wikipedia thesaurus that we generated, where we treat the different relations in the thesaurus according to their different importance, in order to enhance traditional content similarity measure for text clustering. To evaluate the performance of the proposed method, we have performed an empirical evaluation on two real datasets – Reuters and OHSUMED. The experimental results show that with the help of the Wikipedia thesaurus, the clustering performance based on our proposed framework is improved over the previous methods. Moreover, with the optimized weights for hypernym, synonym, and associative concepts tuned with a few labeled data users provided, the clustering performance can be further improved.

The rest of our paper is organized as follows: Section 2 describes the related works. In Section 3, our method of building thesaurus from Wikipedia is discussed. We outline the algorithm that utilizes Wikipedia thesaurus to improve text clustering in Section 4 before introducing our data set and evaluating our algorithm's performance in Section 5.

2. RELATED WORKS

To date, the work on integrating semantic background knowledge into text clustering (classification) or other related tasks is quite few and the results are not good enough. Buenaga Rodriguez et al [16] and Urena Loez et al [24] successfully integrated the WordNet resource for a document categorization task. They improved classification results of Rocchio and Widrow-Hoff algorithms on Reuters corpus. In contrast to our approach, [16] utilize WordNet in a supervised scenario without employing WordNet relations such as hypernyms and associative relations. Meanwhile, they built the term vectors manually. Dave *et al.* [17] has utilized WordNet synsets as features for document representation and subsequent clustering. He did not perform word sense disambiguation and found that WordNet synsets decreased clustering performance in his experiments. Hotho *et al.* [1] integrated WordNet knowledge into text clustering, and investigated word sense disambiguation strategies and feature weighting schema through considering the hypernym relations from WordNet. The experimental results on Reuters corpus show improvements compared with the best baseline. However, considering the few word usage contexts provided by WordNet, the word sense disambiguation effect is quite limited. Meanwhile, the enrichment strategy which appends or replaces document terms with their hypernym and synonym is overly simple.

Gabrilovich *et al.* [2][3] proposed and evaluate a method to render text classification systems with encyclopedic knowledge – Wikipedia and ODP. They first build an auxiliary text classifier that can match documents with the most relevant articles of Wikipedia, and then augment the conventional *BOW* representation with new features which are the concepts (mainly the titles) represented by the relevant Wikipedia articles. Empirical results show that this representation improve text categorization performance across a diverse collection of datasets. However, they do not make full use of the rich relations in Wikipedia such as hyponym, synonyms and associated terms. In addition, they also employ similar document enrichment strategy as [1]. Pu, *et al.* [22] proposed to extract concept relations from Wikipedia and utilize the extracted relations to improve text classification. However, they also treat the hyponym and associative concepts equal with terms in document.

3. WIKIPEDIA THESAURUS

Wikipedia is a dynamic and fast growing resource – articles about newsworthy events are often added within few days of their occurrence [23]. Each article in Wikipedia describes a single topic; its title is a succinct, well-formed phrase that resembles a term in a conventional thesaurus [5]. Meanwhile, each article must belong to at least one category of Wikipedia. Hyperlinks between articles keep many of the same semantic relations as defined in international standard for thesauri [1], such as equivalence relation (synonymy), hierarchical relation (hypernym) and associative relation. However, as an open resource, it inevitable includes much noise. To make it a clean and easy-to-use as a thesaurus, we first preprocess the Wikipedia data to collect Wikipedia concepts, and then explicitly derive relationships between Wikipedia based on the structural knowledge of Wikipedia.

3.1 Wikipedia Concept

Each title of Wikipedia articles describes a topic, and we denote it as a concept. However, some of the titles are meaningless – it is only used for Wikipedia management and administration, such as "1980s", "List of newspapers", .etc. Hence, we first filter Wikipedia titles according to the rules describing below (titles satisfy one of below will be filtered):

- The article belongs to categories related to chronology, i.e. "Years", "Decades" and "Centuries".
- The first letter is not a capital one.
- The title is a single stopword.

- For a multiword title, not all words other than prepositions, determiners, conjunctions, or negations are capitalized.
- The title occurs less than three times in its article.

3.2 Synonymy

Wikipedia guarantees that there is only one article for each concept by using "Redirect" hyperlink to group equivalent concepts to the preferred one. The "Redirect" link copes with capitalization and spelling variations, abbreviations, synonyms, and colloquialisms. Synonymy in Wikipedia mainly comes from these redirect links. In addition, Wikipedia articles often mention other concepts, which already have corresponding articles in Wikipedia. The anchor text on each hyperlink may be different with the title of the linked article. Thus, anchor texts can be used as another source of synonymy.

3.3 Polysemy

Wikipedia contains a lot of disambiguation pages, which are created for ambiguous terms, i.e. terms that denote two or more entities. For example, the term "Puma" may refer to either a kind of animal or a kind of racing car or a famous sportswear brand. Therefore, Wikipedia provides disambiguation pages that present various possible meanings from which users could select articles corresponding to their intended concepts.

3.4 Hypernymy (Hierarchical Relation)

In Wikipedia, both articles and categories can belong to more than one category, i.e. the article of "Puma" belongs to two categories: "Cat stubs" and "Felines". These categories can be further categorized by associating them with one or more parent categories. The category structure of Wikipedia does not form a simple tree-structured taxonomy but a directed acyclic graph, in which multiple categorization schemes co-exist simultaneously [5]. Thus, Wikipedia category system is not taxonomy with a fully-fledged subsumption hierarchy, but only a thematically organized thesaurus. To extract the real "is a" relations from Wikipedia categories, we utilize the methods proposed in [18] to derive generic "is a" relation from category links. Thus, we can get hypernym for each Wikipedia concepts.

3.5 Associative relations

Each Wikipedia article contains a lot of hyperlinks, which express relatedness between them. As Milne et al. [5] mentioned that links often occur between articles that are only tenuously related. For example, comparing the following two links: one from the article "Cougar" to the article "South America", the other from the article "Cougar" to the article "Puma"; it is clear that the former two articles are not as related as the later pair. So, how to measure the relatedness of hyperlinks within articles in Wikipedia is an important issue. Here, we introduce two kinds of measurements to rank links in an article of Wikipedia.

Content based measure

The cosine similarity of article pairs in Wikipedia may reflect the relatedness between the two concepts. However the drawback of this measurement is the same as that of *BOW* approach, since it only considers terms appeared in text documents. We need synthesize other measurements with this one.

Out-linked category based measure

Another method to measure the relatedness between a pair of Wikipedia articles is to compare the similarity between out-linked categories of the two articles. Through observation, we found that if two articles share some out-linked categories, the concepts described in these two articles are most likely related. For example, Table 1 shows part of the common out-linked categories shared by "Data mining", "Machine learning" and "Computer Network". Obliviously, the category distribution between "Data mining" and "Machine learning" is more similar than that between "Data mining" and "Computer Network". In Wikipedia, each article has a lot of hyperlinks which point to other related articles in Wikipedia, and each Wikipedia article belongs to at least one category. Thus, for each Wikipedia article c_i, we can built its out-link category feature vector $f_i = \{cate(c_1), cate(c_2), ...cate(c_k)\}$, $c_1, c_2 ... c_k$ is the out-linked article of c_i ,and $cate(c_k)$ denotes the categories c_k belongs to. Then, we measure the relatedness of two Wikipedia articles c_i and c_j using the cosine similarity of f_i and f_j after weighted with *TFIDF*.

Table 1: Out-link Categories of the article "Data mining" and the article "Machine learning"

Category Name	Data Mining	Machine Learning	Computer Network
computer science	2	6	4
applied mathematics	2	2	0
classification algorithms	5	7	0
Statistics	9	10	0
machine learning	4	14	0
business intelligence	4	2	1
data management	6	0	21
computer networks	1	2	1

Combination of the two measures

To get an overall relatedness of two Wikipedia concepts, we combine the above two measure using the follow equation:

$$S_{Overall} = \lambda_1 \cdot S_{content} + (1-\lambda_1) \cdot S_{olc} \quad (1)$$

Where, λ_1 is the weight parameter to control the influence of content based similarity measure $S_{content}$ and out-linked category based measure S_{olc}.

Then, we ranked all the out-linked concepts for each Wikipedia concept using the above equation, and we denote the out-linked concepts with relatedness above certain threshold (in our experiments, it is set to 0.2) as associative ones for each concept.

4. IMPROVING TEXT CLUSTERING USING WIKIPEDIA THESAURUS

In this section, we first describe the traditional text document similarity measure based on "*BOW*", and previous

text representation enrichment strategies. Then, we introduce our framework which integrates hierarchical, synonym and associative relations from our built Wikipedia thesaurus with traditional text similarity measure.

4.1 Traditional Text Similarity Measure

Intuitively, if two articles address similar topics, it is highly possible that they share lots of substantive terms, while two irrelevant articles are most likely using different vocabulary therefore seldom share any terms. Thus, the text document can be represented as weighted bag of words. After remove the stopwords and stemmed by stemmer such as Porter stemmer [8], the stemmed terms construct a vector representation $\overrightarrow{t_d}$ for each text document. Then, *TFIDF* weighs each term in a document, $\overrightarrow{t_d} = (TFIDF(d,t_1),\ldots, TFIDF(d,t_m)$. Finally, we compute semantic relatedness of a pair of text fragments as the cosine similarity of their corresponding term vectors which is defined as

$$S_{TFIDF} = \frac{\overrightarrow{t_a}\overrightarrow{t_b}}{|\overrightarrow{t_a}||\overrightarrow{t_b}|} \quad (2)$$

4.2 Traditional Text Representation Enrichment Strategies

As introduced in the related works, to break the bottleneck of traditional *"BOW"* representation, previous approaches enriched text representation with external resources such as WordNet and ODP. We summarize their processes as below:

First, they generate new features t_{d-new} for each document in the dataset. The features can be synonym or hypernym for document terms as in [1][16][24], or expanded features for terms, sentences and documents as in [2][3].

Second, the generated new features replace or append to original document representation t_d , and construct new vector representation $\overrightarrow{t_{d-ext}}$ for each text document. After weighted with *TFIDF*, the similarity measure of document pairs is defined as

$$S_{d-ext} = \frac{\overrightarrow{t_{a-ext}}\overrightarrow{t_{b-ext}}}{|\overrightarrow{t_{a-ext}}||\overrightarrow{t_{b-ext}}|} \quad (3)$$

4.3 Our Framework

In this section, we will introduce our framework of leveraging the semantic relations in our built Wikipedia thesaurus to enhance traditional content similarity measure for text clustering.

4.3.1 Mapping Text Documents into Wikipedia Concept Sets

To use Wikipedia thesaurus to enhance clustering, one of the key issues is how to map terms in text documents to Wikipedia concepts. Considering frequently occurred synonymy, polysemy and hypernymy in text documents, accurate allocation of terms in Wikipedia is really critical in the whole clustering process.

To facilitate the mapping process of phrases in text document to Wikipedia concepts, we build a phrase index which includes the phrases of Wikipedia concepts , their synonym, and polysemy in Wikipedia thesaurus. Based on the generated Wikipedia phrases index, all candidate phrases can be recognized in the web page. We use the Forward Maximum Matching algorithm [25] to search candidate phrases, which is a dictionary-based word segmentation approach. It is necessary to do word sense disambiguation to find its most proper meaning mentioned in documents, if a candidate concept is a polysem. Silviu [12] proposed a disambiguation method which augments the Wikipedia category information with Wikipedia pages content, and the implemented system shows high disambiguation accuracy. We adopt Silviu's method to do word sense disambiguation for the polysem concepts in the document.

4.3.2 Enriching Similarity Measure with Hierarchical Relation

In Wikipedia, each concept belongs to one or more categories, while these categories are further belonged to more higher level categories, forming an acyclic category graph. The set of categories contained in the category graph of a given concept c is represented as $\mathrm{Cate(c)} = \{\mathrm{cate_{c1}}, \mathrm{cate_{c2}},.., \mathrm{cate_{cm}}\}$. In the category graph, a category may have several different paths link to a concept. We calculate the distance $\mathrm{dis(c, cate_i)}$ by the length of the shortest path from the concept c to the category $\mathrm{cate_i}$.

Sharing a common category indicates that two articles are somehow related. However, it may take several steps to let two articles find their commonly belonged category. Intuitively, those high level categories have less influence than those low level categories since low level categories are more specific and therefore can depict the articles more accurate. We represent the influence of categories on γ^{th} layer on concept c as $\mathrm{Inf_\gamma(c)}$ and define $\mathrm{Inf_1(c)} = 1$. A decay factor $\mu \in [0,1]$ is introduced for higher levels of categories. Therefore, we have $\mathrm{Inf_\gamma(c)} = \mu\mathrm{Inf_{\gamma-1}(c)} = \mu^{\gamma-1}\mathrm{Inf_1(c)}$. As each Wikipedia concepts has more than one category, and each category has more than one parent categories, a big γ will introduce too many categories. Therefore, we set $\gamma \leq 3$in our experiments. Thus, for each concept c , we can build a category vector $\overrightarrow{\mathrm{cate_c}} = \{\mathrm{Inf(c, cate_{c1})}, \mathrm{Inf(c, cate_{c2})},.., \mathrm{Inf(c, cate_{cm})}\}$, where $\mathrm{Inf(c, cate_{ci})} = \mathrm{Inf_{dis\,(c,cate\ ci\ )}(c)}$ which indicates the influence of category $\mathrm{cate_{ci}}$ on concept c. For the collection C which contains all the concepts in document d , the corresponding category vector can be calculated as $\overrightarrow{\mathrm{Cate_c}} = \sum_{\mathrm{ci}\in C}\overrightarrow{\mathrm{cate_c}}$ and the similarity measure using category vectors of document is defined as:

$$S_{cate} = \frac{\overrightarrow{Cate_a}\overrightarrow{Cate_b}}{|\overrightarrow{Cate_a}||\overrightarrow{Cate_b}|} \quad (4)$$

Considering the original document content, the similarity measure S' can be represented as:

$$S' = (1-\alpha)S_{TFIDF} + \alpha S_{cate} \quad (5)$$

where α is used to control the importance of S_{cate} in the combined measure. Specifically, when use the category decay factor μ, Eq. 5 can be rewritten as

$$S' = (1-\alpha)S_{TFIDF} + \alpha S_{cate,\mu} \quad (6)$$

4.3.3 Enriching Similarity Measure with Synonym and Associative Relation

To better relieve *BOW* shortcomings, synonym and associative relations in Wikipedia can be used to include

more related concepts in the similarity measure. For each concept c_i in Wikipedia, a set of related concepts $rela(c_i) = ((c_{r1}, w_{r1}), (c_{r2}, w_{r2}), .., (c_{rn}, w_{rk}))$ are selected from its synonym and associative concepts, in which c_{ri} is the ith related concepts of c_i and w_{ri} is the relatedness between c_i and c_{ri} – For synonym $w_{ri} = 1$, and for associative concepts w_{ri} is measured by Equation 1. Given two articles, we use the two weighted sets of concepts to measure their similarity. Consider two sets of concepts represent as $C_a = \{(c_{a1}, f_{a1}), (c_{a2}, f_{a2}), .., (c_{an}, f_{an})\}$ and $C_b = \{(c_{b1}, f_{b1}), (c_{b2}, f_{b2}), .., (c_{bm}, f_{bm})\}$, where c_{ak} $(0 < k \leq n)$ and $c_{bj} (0 < j \leq m)$ is Wikipedia concept in the two articles, and f_{ak}, f_{bj} are their corresponding weights. We expand C_b with all the related concepts of its elements c_{bj} that are contained in C_a. The expanded weighted concept set is defined as:

$$C_{ext} = \{(c, w_c) | c \in C_a \wedge \exists c_{bi} \in C_b \; s.t. \; c \in rela(c_{bi}))\} \quad (7)$$

where w_c is calculated by summing up all weighted occurrence of corresponding related concepts. We append the expanded concept set C_{ext} to C_b and get the extended C_b as:

$$C_{b-ext} = C_b + C_{ext} \quad (8)$$

Given two concepts sets, we always choose the smaller one as C_b since the expanding procedure always makes the set bigger. And we define the similarity as:

$$S_{asso} = \frac{C_a C_{b-ext}}{|C_a||C_{b-ext}|} \quad (9)$$

We give an example of two concepts sets $C_a = \{(CS, 1), (ML, 1)\}$ and $C_b = \{(DM, 1), (DB, 1)\}$ (CS – Computer Science, ML – Machine Learning, DM – Data Mining, DB - Database). We gave the similarity measure of the four concepts in Table 3. We can get the extended set $C_{ext} = \{(CS, 0.3 + 0.3), (ML, 0.7 + 0.1\} = \{(CS, 0.6), (ML, 0.8)\}$ and $C_{b-ext} = \{(DM, 1), (DB, 1), (CS, 0.6), (ML, 0.8)\}$. The similarity between C_a and C_b is 0 since they share no common concepts. However, using C_{b-ext}, we get $S_{simi} = 0.57$ indicating there is correlation between C_a and C_b.

Table 2: The similarity table of four selected concepts

	Computer Science	Machine Learning
Data Mining	0.3	0.7
Database	0.3	0.1

Considering the original document content, the combined similarity measure is defined as:

$$S'' = (1 - \beta) S_{TFIDF} + \beta S_{asso} \quad (10)$$

where β is used to control the importance of S_{asso} in the combined measure.

4.3.4 The Combination

In the previous sections, we describe the methods to combine Wikipedia hierarchical relation, synonym and associative relation with traditional text document similarity measure. In this section, we incorporate both of them into the similarity measure using a linear combination and it is defined as:

$$S_{\text{Comb}} = (1 - \alpha - \beta) S_{TFIDF} + \alpha S_{cate,\mu} + \beta S_{asso} \quad (11)$$

where α and β weight the importance of hierarchical relation, synonym & associative relation in the similarity measure, respectively. As text clustering is an unsupervised method, where we do not have labeled data, we cannot tune the parameters with validation data. Thus, in these cases, we can set α and β to equal weights ($\alpha = \beta = 1/3$). If users can provide some prior-knowledge or validation data which specifies that some documents be clustered together, the weights for α,β can be optimized based on such data.

5. Experiments

Our incorporation of Wikipedia background knowledge is independent of the concrete clustering methods. The only requirement we have is that the algorithm could achieve good clustering results in an efficient. K-Means [21] is a widely-used clustering algorithm with good accuracy and efficiency. In our experiments, we use K-Means clustering algorithm to evaluate our proposed methods, and we will try to use other clustering algorithms in our future work.

5.1 Wikipedia Data

As an open source project, Wikipedia content is easily obtainable through downloading from http://download.wikipedia.org. It is available in the form of database dumps that are released periodically. The version we used in our experiments was released on Sep. 9, 2007. We identified over four million distinct entities (articles and redirections) that constitute the vocabulary of thesaurus. These were organized into 127,325 categories with an average of two subcategories and 26 articles each. The articles themselves are highly inter-linked; each links to an average of 25 others. After filtering Wikipedia concepts as described in Sec 3.1, we got 1,614,132 concepts.

5.2 Clustering Data Set

Reuters-21578[10] is a news corpus containing 11,367 manually labeled documents classified into 82 clusters, with 9494 documents uniquely labeled. We filter those clusters with less than 15 documents or more than 200 documents, leaving 30 clusters comprising of 1,658 documents.

OHSUMED[11] is a subset of MEDLINE containing 348,566 medical documents, about two-thirds of which (233,445) also have an abstract. We choose a subset of 18,302 abstracts which are classified into 23 categories, with each category contains from 56 to 2,876 abstracts.

5.3 Evaluation Criteria

The purity measure based on precision measure in information retrieval field is applied to evaluate the performance of our strategies. Both purity and inverse purity are calculated to collaboratively depict the accuracy of the clustering result. Given a data set D, $M = M_1, M_2, .., M_n$ represent the n manually labeled clusters, and $C = C_1, C_2, .., C_n$ represent the n clusters generated using our algorithm.

For each $C_i \in C$ and $M_j \in M$, precision of C_i and M_j is defined as:

$$Pre(C_i, M_j) = \frac{|C_i \cap M_j|}{|C_i|}. \quad (12)$$

The purity of the clustering result is defined as:

$$Pur(C, M) = \sum_{C_i \in C} \frac{|C_i|}{|C|} \max_{M_j \in M} Pre(C_i, M_j) \quad (13)$$

and the corresponding inverse purity is defined as

$$IPur(C,M) = \sum_{M_i \in M} \frac{|M_i|}{|M|} \max_{C_j \in C} Pre(M_i, C_j). \quad (14)$$

Actually, we can see that

$$IPur(C,M) = Pur(M,C), \quad (15)$$

They collaboratively measure the accuracy of the clustering result. Though purity and inverse purity are often positively correlated, they do not always get their peak at the same point. In this case, we choose the point with the highest sum of purity and inverse purity as the global optimal.

5.4 Experimental Results

In our experiments, each evaluation result described in the following denotes an average from 10 test runs performed on given corpus for a given combination of parameters with randomly chosen initial values for K-Means. We also applied t-tests to check for significance with a confidence of 99%.

We used three baselines for comparing: The first one is K-Means clustering with traditional text document similarity measure – we denote it as *BASE1*; The second one is K-Means clustering with document representation improved with Gabrilovich's feature generation technique on Wikipedia [2] – we denote it as *BASE2*; The third one is K-Means clustering with Hotho's document representation enrichment with WordNet [1] – we denote it as *BASE3*. In this section, we first introduce the overall experimental performance.

5.4.1 Overall Performance

We conducted several experiments: *BASE1*, *BASE2*, *BASE3*, transitional text similarity measure with hierarchical relations (*HR*), transitional text similarity measure relations with synonym and associative relations (*SAR*), and the combination of *HR* and *SAR* (*COB*). For *BASE1*, *BASE2* and *BASE3*, We select parameters according to their best performance setting in our experiments. For *HR*, *SAR*, and *COB*, we fixed several parameters for the rest experiments. i.e. λ_1 is set to 0.5 in associative concepts ranking of Wikipedia thesaurus, and $\mu = 0.5$. As it is suppose to be no validation data for document clustering, we use the average weight for α, and β on both Reuters and OHSUMED datasets (e.g. for *COB* $\alpha = 1/3$, $\beta = 1/3$). The experimental performance of these six methods is summarized in Table 3. *Purity* and *Inverse* in the table represent the purity and inverse purity measures for each method and *Impr* represents the average improvement of purity and impurity measures compared with *BASE1*.

Table 3: Baseline,HR, SAR and COB performance results

	Reuters			OHSUMED		
	Purity	Inverse	Impr	Purity	Inverse	Impr
BASE1	0.603	0.544		0.414	0.343	
BASE2	0.605	0.548	0.53%	0.427	0.354	3.17%
BASE3	0.607	0.556	1.43%	0.435	0.358	4.72%
HR	0.604	0.547	0.36%	**0.459**	**0.388**	**12.0%**
SAR	0.652	0.593	8.57%	0.438	0.359	5.23%
COB	**0.655**	**0.598**	**9.28%**	0.449	0.381	9.77%

From the performance results in Table 3, we can see that within the three baseline algorithms, both Gabrilovich's feature generation and Hotho's WordNet enrichment can improve clustering performance compared with traditional *BOW* representation. However, Gabrilovich's method can only get 0.53% and 3.17% improvement on Reuters and OHSUMED compared with *BASE1*, which is less than Hotho's *BASE3* (1.43% and 4.72%) improvement and far less than our *COB* method (9.28% and 9.77%) improvement. As we use the same weights for hierarchical relation and synonym & associative relation, the result indicates that Wikipedia thesaurus can improve text clustering performance.

Comparing our proposed HR, SAR, and COB experiment results, we find that hierarchical relation, synonym & associative relation and their combination improve clustering performance on both Reuters and OHSUMED data sets - (0.36%, 8.57%, 9.28%) vs (12%, 5.23%, 9.77%) improvement. Reuters benefits from combing both relations – 9.28% improvements in *COB*, in which synonym & associative relation plays a more importance role than hierarchical relations (HR only gets 0.36% improvement). However, for OHSUMED, hierarchical category relations has already improved the performance a lot (12% improvements) so that adding synonym and associative relations do not offer more help in the *COB* experiment – the improvement is even decreased from 12% to 9.77%. More specifically, we can see that adding synonym and associative relations contributes more in the clustering of the Reuters data set while adding hierarchical relations are more helpful in the OHSUMED data. This can been explained from the fact that OHSUMED are professional medicine articles that the implicit hyponymy can be extracted only through hierarchical category relations of Wikipedia, while Reuters are more general news articles often varying words and phrases therefore adding related concepts is more helpful in this case. It also indicates that for different kinds of datasets, the effect of the hierarchical relation and synonym & associative relation is various, and their combination can take advantages their both generally.

5.4.2 Optimizing Parameters α and β

In previous experiments, we give the same fixed values for α and β, which means we treat hyponym and associative concepts are equal important. However, as shown in Table 3, for different datasets, when computing the similarity between document pairs, the original document content, hyponym and associative concepts should have different importance. As we know, clustering is unsupervised, and it does not need labeled training data. But it does not mean that users should not provide prior-knowledge or validation data which specifies that some documents be clustered together. Although it is laborious for users to label a large amount of data, it is worthwhile if the clustering performance can be greatly improved when users can provide some labeled data for optimizing some important parameters. In the following experiments, we will evaluate the clustering performance when using different amount of labeled data to optimize parameters α and β.

Figure 1 and 2 show the purity and inverse purity curve of our algorithm under different number of labeled validation

data on Reuters and OHSUMED. The Axis-x of the two figures denotes the percentage of documents with label in Reuters or OHSUMED used for optimizing parameters α and β . From the two figures, we can see that, as we expected, with the increase of labeled tuning documents, the purity and inverse purity increase gradually. Especially when the number of labeled tuning data increases from 0 to 5%, the clustering performance improved a lot – On OHSUMED, purity (inverse purity) increases from 0.449 to 0.467 (0.381 to 0.392), which means the clustering performance can be greatly improved if users can provide a small amount of labeled data for tuning parameters α and β. When the tuning data increases from 10% to 20%, the curve of purity and inverse purity still goes up, but not as quick as the one from 0 to 5%. If we still increase the labeling data (from 20% to 50%), the curve becomes quite stable, which indicates that 20% of data is sufficient for tuning optimal weights for hierarchical and associative relations. Table 4 summarizes the best results we get on Reuters and OHSUMED after using optimized weights. As shown in Table 4, based on the optimal weights, we can get 0.697 purity and 0.636 inverse purity on Reuters, and 0.485 purity 0.414 inverse purity on OHSUMED.

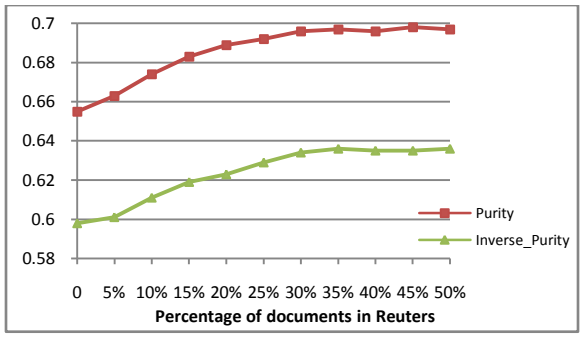

Figure 1: Impact of tuning document number in Reuters for clustering performance

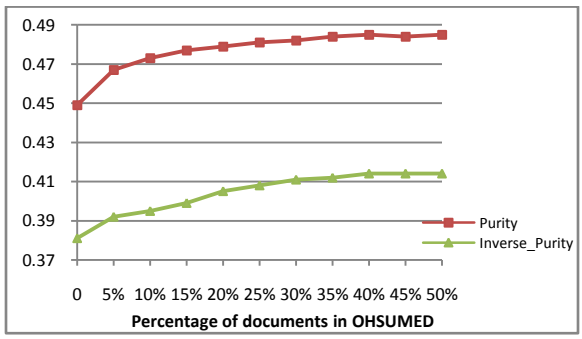

Figure 2: Impact of tuning document number in OHSUMED for clustering performance

Table 4: Clustering performance using optimized weights

	Reuters			OHSUMED		
	Purity	Inverse	Imp	Purity	Inverse	Impr
BASE1	0.603	0.544		0.414	0.343	
COB(optimized)	**0.697**	**0.636**	**16.2%**	**0.485**	**0.414**	**18.8%**

5.4.3 Detailed Analysis

To better understand the reasons why our approach works better than Gabrilovich's feature generation and Hotho's WordNet enrichment, we analyze the generated features of the three methods for the Reuters document #15264, and summarize part of their generated features in Table 5. The left part of "->" is the original word (term or phrase) in the document used for generating new features, and the right part of it is the generated new features (*S* denotes Synonym, *H* denotes Hypernym, and *A* denotes associative terms). Comparing the generated features of Hotho's WordNet enrichment and our Wikipedia based approach in Table 5, we find that words used for generating features in Hotho's method are single terms and most of them are polysemy; While our approach generates features not only for single terms but also for phrases, and most of them are name entities not covered by WordNet. Due to rich context provided by Wikipedia, our disambiguation module can find the proper meaning for most of polysemy in document – TECK can be mapped to Teck Cominco - a mining company in Canada, which is directly related to the topic of the document (mining). Although Gabrilovich also utilizes Wikipedia to generate features, the classification-based feature generation approach brings a lot of noise features, and it also doesn't contain hypernym features which are proven quite useful in finding sharing topics between document pairs.

Table 5: Generated features of Hotho's, Gabrilovich's and our method.

Hotho	copper->CO(S);metallic element(H);metal(H);conductor(H);cupric(A) venture->undertaking(H);project(H);task(H);venturer(A) highland->elevation(H) valley->natural depression(H);depression(H) british->brits(S);nation(H);land(H);country(H) columbia->nation(H);country(H) affiliate->associate(H);affiliation(A) mining->production(H);mine(A);excavate(A) negotiation->discussion(H);word(H);negotiate(A) complete->end(H);terminate(H);completion(A); finish(A) administration->management(H);direction(H);administer(A) reply->response(S);statement(H);answer(A) silver->Ag(S);noble metal(H);conductor(H) ounces->ounce(S);troy ounce(S);troy unit(H); unit(A) Molybdenum->Mo(S);metallic element(H);metal(H)
Gabri	Teck; John Townson; Cominco Arena;Allegheny Lacrosse Officials Association;Scottish Highlands;Productivity;Tumbler Ridge, British Columbia;Highland High School;Economy of Manchukuo;Silver;Gold (color);Copper (color);
Ours	TECK -> Teck Cominco(S);Mining companies of Canada(H) british columbia->british columbian(S);provinces and territories of canada(H);greater vancouver regional district(H) teck cominco-> Mining companies of Canada(H)con mine(A) mining->miner(S);metal mining(S);industries(H);resource extraction(H);mining engineering(A) molybdenum->element 42(S);chemical elements(H);dietary minerals(H);refractory metals(A) joint venture->strategic alliance(H);joint ventures(H);joint venture broker(A);shell-mex and bp(A) copper->Copper(H);Chemical elements(H);dietary minerals(H); chemical element(A);ductile metal(A)

5.4.4 Parameter Setting

As mentioned in Sec. 3.5, we adopt two ways to measure the associative relatedness between Wikipedia concepts. Here we introduce the method to tune λ_1 of Equation 1. First we select 10 Wikipedia concepts randomly, and then

extract all the out-linked concepts in the Wikipedia articles corresponding to the 10 concepts. To obtain high quality ground truth for tuning, we asked three assessors to manually label all the linked concepts in the 10 articles to three relevance levels (relevant - 3, neutral - 2, and not relevant - 1). The labeling process was carried out independently among assessors who are graduate students and have good command of the English. No one among the three assessors could access the labeling results of others. After labeling, each out-linked concept in the 10 articles is labeled with 3 relevance tags, and we use average value as the final relatedness value. Based on the labeled data, we calculate *TFIDF* similarity and out-linked category similarity between the 10 concepts and their out-linked concepts. We tune the value of λ_1 from 0.1, 0.2, up to 1.0, and thus we can find a proper value of λ_1, with which the result of linear combination matches the user evaluation result best. From experiments, λ_1 is set to 0.5. Other parameter that we pay special attention to category decay factor μ we used in similarity measure combining hierarchical relations. We conduct extensive experiments on different parameter settings and $u = 0.5$ always show the best results. Therefore, we set $u = 0.5$ in our experiments.

6. Conclusion and Future Works

Wikipedia is a huge resource of encyclopedia knowledge which contains a lot of name entities that are widely used in our daily life. But it is not structured as WordNet and cannot be used for other application directly. Therefore, we first proposed a way to mine synonym, hypernym and associative relations explicitly for each concept through analyzing the rich links in Wikipedia, and build it as an easy-to-use thesaurus. Then, we introduce a framework to integrate the hierarchical, synonym, and associative relations in built Wikipedia thesaurus to traditional text similarity measure to facilitate document clustering. The text clustering experiments on two datasets indicate that with the help of our built Wikipedia thesaurus, the clustering performance of our method is improved compared with previous methods. Meanwhile, with the optimized parameters based on a few labeled data users provide, the clustering performance can be further improved - 16.2% and 18.8 improvement compared with the baseline on Reuters and OHSUMED, respectively. For the future work, as Wikipedia is multilingual encyclopedia; it provides more than 20 languages with rich inter-links between them. Thus, we can use the multilingual relations to explore the application in Cross-language Information Retrieval and Cross-language Text Categorization, which is our next step for future work.

7. REFERENCES

[1] A. Hotho, S. Staab and G. Stumme. Wordnet improves text document clustering. *In Proceedings of the Semantic Web Workshop at SIGIR'03*

[2] E. Gabrilovich and S. Markovitch. Feature Generation for Text Categorization Using World Knowledge. *In IJCAI' 05.*

[3] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. *AAAI'06.*

[4] E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *IJCAI' 07.*

[5] D. Milne, O. Medelyan and I. H. Witten. Mining Domain-Specific Thesauri from Wikipedia: A case study. In *WI'06*.

[6] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL06*.

[7] M. Strube and S. P. Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI'06*.

[8] M. F. Porter. An algorithm for suffix stripping. Program, 14(3). 1980. pp. 130–137.

[9] E. Agirre and G. Rigau. A Proposal for Word Sense Disambiguation using Conceptual Distance. In the *Proceedings of the First International Conference on Recent Advances in NLP*. 1995.

[10] Reuters-21578 text categorization test collection, Distribution 1.0. Reuters. http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[11] W. Hersh, C. Buckley, T. Leone and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *SIGIR'94*, pp. 192–201.

[12] K. Lang. Newsweeder: Learning to filter netnews. In *ICML'95*, pp. 331–339.

[13] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML'98*, pp. 137–142.

[14] F. Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys 34(1). 2002.

[15] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR'99*, pp. 42–49.

[16] M. de Buenaga Rodrıguez, J. M. G. Hidalgo, and B. Dıaz-Agudo. Using WordNet to complement training information in text categorization. In *Recent Advances in Natural Language Processing II*, volume 189. 2000.

[17] D. M. P. Kushal Dave, Steve Lawrence. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW'03*.

[18] Ponzetto, Simone Paolo; Strube, Michael (2007). Deriving a Large Scale Taxonomy from Wikipedia In: *AAAI '07*, pp.1440-1445

[19] Miller, G. (1995). WordNet: A lexical database for english. CACM, 38, 39–41.

[20] Open Directory Project, http://dmoz.org

[21] Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *KDD Workshop on Text Mining*.

[22] Pu Wang, Jian Hu, .etc. Improving test categorization by using Encyclopedia knowledge, In: *ICDM'07*.

[23] Wikipedia, http://en.wikipedia.org/wiki/Wikipedia:About

[24] Ure˜na L´oez, M., & Hidalgo, J. M. G. (2001). Integrating linguistic resources in tc through wsd. *Computers and the Humanities*, 35(2), 215–230

[25] Wong, P. and Chan, C., Chinese word segmentation based on maximum matching and word binding force. in *Proc. of the 16th conference on Computational linguistics*, 1996.