Improving Text Classification by Using Encyclopedia Knowledge[*]

Pu Wang[1], Jian Hu[2], Hua-Jun Zeng[2], Lijun Chen[1], Zheng Chen[2]

[1]Department of Computer Science
Peking University, Beijing 100871, P.R. China
{pwang,ljchen}@db.pku.edu.cn

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

Abstract

The exponential growth of text documents available on the Internet has created an urgent need for accurate, fast, and general purpose text classification algorithms. However, the "bag of words" representation used for these classification methods is often unsatisfactory as it ignores relationships between important terms that do not co-occur literally. In order to deal with this problem, we integrate background knowledge - in our application: Wikipedia - into the process of classifying text documents. The experimental evaluation on Reuters newsfeeds and several other corpus shows that our classification results with encyclopedia knowledge are much better than the baseline "bag of words" methods.

1. Introduction

Traditionally, document classification has been based on a variation of the "bag of words" (*BOW*) approach, which represents the features in a document by the weighted occurrence frequencies of individual word in it. Apparently, the *BOW* approach is limited since it can only use the set of terms explicitly mentioned in the documents and ignores relationships between important terms that do not co-occur literally, e.g. terms as "Puma" and "Cougar" are concepts belonging to the same genus "Felines", but they are not equal literally; so *BOW* approach doesn't work at this circumstance. To overcome this kind of problem, several research works have been done to exploit ontology for content-based categorization of large document corpus. Hotho et al [1] utilizes *WordNet*, structured term ontology, to improve the *BOW* text representation. It adopts strategies which represent text documents with taking background knowledge into account to various extents. For instance, terms like "Puma" and "Cougar" are considered to be similar, because they both are subconcepts of the concept "Felines" in *WordNet*. *BOW*, however, treats these two words as totally difference ones. Although experiment results showed some improvement in classification or clustering accuracy, *WordNet* is manually constructed, its coverage is far too limited and its maintenance is painstaking. Therefore other research works take advantages of the world knowledge bases, whose coverage are more extensive than *Wordnet*, such as Open Directory Project (*ODP*) and *Wikipedia* – the largest human encyclopedia to date: Gabrilovich et al [2] applied feature generation techniques to text processing based on *ODP*; Gabrilovich et al [3] empowered machine learning techniques to *Wikipedia* to enrich document representation. Many text classification experiments have confirmed better performance of the improved categorization methods, which classify documents with the help of backgrougd-knowledge-based features generated from *ODP* or *Wikipedia*, than that of *BOW* approach.

However, since *ODP* and *Wikipedia* are not structured thesauri as *Wordnet*, when enriching documents with features generated by *ODP* or *Wikipedia*, they are not as suitable in themselves as *Wordnet* to handle the problems of synonymy and polysemy, which are two fundamental problems in text categorization. To overcome these problems, Gabrilovich et al [2][3] perform feature generation using a multi-resolution approach: Features are generated for each document at the level of individual words, sentence, paragraph, and finally the entire document. This feature generation procedure acts similar to a retrieval process: it receives a text fragment (such as words, sentence, paragraph, or whole document) as input, and then maps it to the most relevant *ODP* categories or *Wikipedia* articles. This method, however, only leverages text similarity between text fragments and *Wikipedia* articles, ignoring the abundant structural information within *Wikipedia* as links. Then, the relevant *ODP* categories' name or *Wikipedia* articles' title retrieved by the former step are treated as new features to enrich document representation. Gabrilovich et al claim that their feature generation method implicitly performs words sense disambiguation: polysemous words within the context of a text fragment are mapped to the concepts which correspond to the sense shared by other context words. But, the processing effort of Gabrilovich's method is too huge, since it has to scan each document many

[*] The work was conducted and completed while the first author was doing internship at Microsoft Research Asia.

1550-4786/07 $25.00 © 2007 IEEE
DOI 10.1109/ICDM.2007.77

times. And based the principle of Information Retrieval, the feature generation procedure inevitably brings a lot of noise, because an article which contains part of input text doesn't mean that it is relevant to the input text. Thus Gabrilovich's method has to rely too much on feature selection to identify correct concepts and eliminate spurious ones, and too much noise makes feature selection less discriminating. Meanwhile, the implicit word sense disambiguation processing can only work at some cases (We will explain this in Sec. 4).

In this work, firstly, we propose a way to build informative encyclopedia thesaurus based on *Wikipedia*, which explicitly derives relationships between concepts of *Wikipedia*, including synonymy, polysemy, hyponymy and associative relation. So our thesaurus takes full advantages of the profuse structural knowledge of *Wikipedia*. The thesaurus serves as a controlled vocabulary that bridge the variety of idiolects and terminologies present in document corpus; and it may facilitate integration world knowledge in *Wikipedia* into text documents, since it resolves synonyms and introduces more general concepts which help identify related topics between text documents. Meanwhile, the thesaurus surpasses any manually constructed thesaurus like *Wordnet* in its coverage, and rivals them in its accuracy. Although Milne et al [5] have built a professional thesaurus of agriculture from *Wikipedia*, it is a domain-specific one, and takes little use of the rich relations within *Wikipedia* articles. However, the thesaurus we built is a more general one, which supports documents of extensive topics, not limited to any field. Beyond that, we then investigate a way to utilize our thesaurus, to facilitate integrating the rich semantic relations between *Wikipedia* concepts to text document represent. Meanwhile, when enriching documents, we try to expand only the most relevant concepts into documents and make explicit word sense disambiguation. Hence, our method won't bring as much noise as [3] does, and makes feature selection more contributing. To come up with what beneficial effects can be achieved with this method, we have performed an empirical evaluation. We compared a simple baseline with different strategies of enriching text documents with *Wikipedia* knowledge to various extents.

The organization of our paper is as follows: Section 2 describes the related works. In Section 3, our method of building thesaurus from *Wikipedia* is discussed. We outline the algorithm of categorization with document enrichment 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 classification or other related tasks is quite few and the results are not good enough even worse. Buenaga Rodriguez et al. [16] and Urena Loez et al. [17] successfully integrate the *WordNet* resource for a document categorization task. They evaluate their methods on Reuters corpus, and show improved classification results of Rocchio and Widrow-Hoff algorithms. In contrast to our approach, [16] and [17] utilize *WordNet* to a supervised scenario, and do not use *WordNet* relations such as hypernyms, not mention associate terms we used in *Wikipedia*. Meanwhile, they build 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] intergrate *WordNet* knowledge into text clustering, and investigate word sense disambiguation strategies and feature weighting schema by considering the hypernym relations from *WordNet*. The experimental evaluation on Reuters shows improvement compared with the best baseline. However, considering the few word usage contexts provided by *WordNet*, the word sense disambiguation effect is quite limited. Meanwhile, *WordNet* does not provide the associate terms as *Wikipedia*.

Gabrilovich et al. [3] propose and evaluate a method to render text classification systems with encyclopedic knowledge – *Wikipedia*. 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 Wikipeida 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 information in *Wikipedia*. *Wikipedia* is not merely a simple article collection; each article describes a single concept: its title is a succinct, well-formed phrase, and hyperlinks between articles capture many of semantic relations [6] such as hyponym, synonyms and associated terms. In addition, as described in [2], the feature generation process will bring much noise although the feature selection step can eliminate some extraneous features.

3. Wikipedia

Launched in 2001, *Wikipedia* is a multilingual, web-based, free content encyclopedia written collaboratively by more than 10,000 regular editing contributors; its articles can be edited by anyone with access to its web site. *Wikipedia* is a very dynamic and quickly growing resource – articles about newsworthy events are often added within days of their occurrence. Today it ranks among one of the most-visited worldwide websites. The ways *Wikipedia* organizes its content are quite similar as the structure of traditional thesauri like *WordNet*, etc.

3.1. Wikipedia as a thesaurus

Wikipedia is the largest encyclopedia in the world. The English version, as of November 30, 2006, contains 4,197,766 articles with about 100 million internal hyperlinks. And 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 the international standard for thesauri [1], such as equivalence relation (synonymy), hierarchical relation (hyponym) and associative relation.

So we treat each topic described in a *Wikipedia* article as a concept and identify several relations between concepts from *Wikipedia*'s structure as described below. Such concepts and relations are just the building blocks of a thesaurus.

3.1.1 Synonymy

As we mentioned above, in *Wikipedia*, each concept is named by a "preferred term", and *Wikipedia* ensures that there is only one article for each concept by using "Redirect" hyperlink to group equivalent concepts to the preferred one. A redirect page which only contains a redirect link exists for each alternative name of the concept that can be used to refer to the preferred one in *Wikipedia*. So synonymy in *Wikipedia* comes from redirect pages. For example, "Ford puma racing" is the full name of the Ford Puma car. Therefore "Ford puma racing" is an alternative name for "Ford puma"; consequently, in *Wikipedia* the article of the title "Ford puma racing" links to the article "Ford puma". And that "Redirect" link also copes with capitalization and spelling variations, abbreviations, synonyms, colloquialisms, and scientific terms. As an instance in [6], an example entry with a considerably higher number of redirect pages is "United States". Its redirect pages correspond to acronyms (U.S.A., U.S., USA, US), Spanish translations (Los Estados Unidos, Estados Unidos), misspellings (Untied States) or synonyms (Yankee land).

In addition, *Wikipedia* articles often mention of other concepts, each already has a corresponding article in *Wikipedia*. And for each mentioned concept in it, a *Wikipedia* article usually links at least its first mention to the corresponding article by using hyperlink. The anchor text on each hyperlink may be different with the title of the linked article. Thus anchor texts are also synonymies of the concepts of linked articles.

3.1.2 Polysemy

Another useful structure in Wikipeida is disambiguation pages, which are created for ambiguous terms, i.e. terms that denote two or more entities. Such as, the term "Puma" may refer to either a kind of animal or a kind of racing car or a famous sportswear brand. So, in *Wikipedia*, it provides disambiguation pages that present various possible meanings from which users could select articles corresponding to their intended concepts. For example, the disambiguation page for the term "Puma" lists 22 associated concepts, from persons, vehicles to sport clubs.

In disambiguation pages, parenthetical expression of each ambiguous term may also help users find intended meaning, i.e. the term "Puma" yields several options, including "Puma", a genus of large cats like Cougar and Jaguarundi, "Puma (car)", a brand of Brazilian-made sports cars, and "PUMA AG", a German shoe and sportswear company. Meanwhile, *Wikipedia* articles correspond to each meaning of an ambiguous term serve as detailed scope notes, since they fully describe the intended meaning of the term.

3.1.3 Hyponymy

The hierarchical organization of *Wikipedia* is shown in its categorization structure. For example, the article about animal "Puma" has corresponding category "Felines", which contains several more specific subcategories and articles, such as "Cats" and "Cougar".

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. So, 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, for a concept in *Wikipedia*, we use the name of each ancestor category as its hyponymy.

3.1.4 Associative relations

Each *Wikipedia* article contains a lot of hyperlinks, which express relatedness between them. For example, there are hyperlinks from the article titled "Puma" to the other articles such as "Felidae", "Cougar" and "Jaguarundi"; some of the linked articles link back to the original one, as the article "Cougar" links back to the article "Puma".

As Milne et al [5] mentioned, 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 closely related as the later pair. So, how to measure the relatedness of hyperlinks within articles in *Wikipedia* is an important issue. Milne et al [5] only considers mutual cross links between articles as intended ones, casting away all other one-way links. However, their method is far too strict. Too many one-way links are discarded, for there are only 2,366,472 mutual links of total 13,947,302 hyperlinks in *Wikipedia*[1]. Meanwhile we found that a lot of one-way links are not tenuously related, and they should be retained. Here's an example: there is only one-way link from article "Data Mining" to the article "Machine Learning", no back link; and we know that "Data Mining" and "Machine Learning" are two closely related concepts; therefore the one-way links between them should be considered. Thus, it is important to evaluate the relatedness of each hyperlink between *Wikipedia* articles and find out the most relevant ones for each article. Here, we introduce three kinds of measurements to rank links in an article of *Wikipedia.*

Content based measure

This measurement is based on vector space model. Relatedness of two linked articles is evaluated by their contained terms: given the occurrence probability of each term in a corpus, the relatedness of two articles is modeled as the extent to which they share terms. Intuitively, if two text documents address to a similar topic, it is inevitable that these two documents may share some common substantive terms; whereas it is not possible that two irrelevant documents contain a lot of same words. Thus a text fragment could be represented as a term vector using *TFIDF* scheme.

[1] This link count statistic is calculated from the snapshot of Wikipedia on Nov. 30, 2006.

To compute semantic relatedness of a pair of text fragments is just to compute the cosine similarity of their corresponding term vectors. Accordingly, the cosine similarity of *TFIDF* may reflect the relatedness of any pair of articles in *Wikipedia*. However the drawback of this measurement is the same as that of *BOW* approach, since they only consider terms appeared in text documents. We need synthesize other measurements together with this one.

Out-link Category based measure

Another method to measure the relatedness of a hyperlink between a pair of *Wikipedia* articles is to compare out-linked categories of the two linked articles. Out-linked categories of an article are the categories that out-linked articles of the original article belong to. In *Wikipedia*, each article must belong to at least one category, and we found that if most of the out-linked categories of two articles focus on several same ones, the concepts described in these two articles are most likely strongly related. As for three related concepts, "Data mining", "Machine learning" and "Computer Network", 75 out linked articles from the article "Machine learning" and 52 out linked ones from the article "Data mining" share 22 same categories; 24 out linked articles from the article "Data mining" and 39 out linked ones from the article "Computer Network" share 10 same categories; 23 out linked articles from the article "Machine learning" and 20 out linked ones from the article "Computer Network" share 10 same categories. Table 1 shows part of the common out-linked categories shared by " Data mining", "Machine learning" and "Computer Network"

Category Name	*Data Mining*	*Machine Learning*	*Computer Network*
information technology	2	3	1
artificial intelligence	2	6	0
computer science	2	6	4
applied mathematics	2	2	0
classification algorithms	5	7	0
artificial intelligence researchers	2	2	0
neural networks	1	3	0
Statistics	9	10	0
information technology management	3	2	5
machine learning	4	14	0
business intelligence	4	2	1
data management	6	0	21
computer networks	1	2	1
Networks	1	3	3
intelligent document	3	0	1

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

So, we first build out-linked category name vector for each *Wikipedia* article. Suppose there are *f* out-linked articles belonging to category c, we denote it as: olc(c,*f*). The out-link category name vector for each article is $\vec{c} = \{\mathrm{olc}(c_1, f_1), \ldots, \mathrm{olc}(c_n, f_n)\}$, which is also a kind of vector, each entry in $\vec{c}$ is a category name. Then we define out-linked category similarity to measure the relatedness of two linked articles. Since $\vec{c}$ is a vector, computing cosine similarity between out-linked category name vectors is applicable. So, the out-link category similarity S_{olc} of two linked articles is defined as:

$$S_{olc} = \frac{\vec{c}_1 \cdot \vec{c}_2}{\vec{c}_1 \times \vec{c}_2} \qquad (1)$$

Where $\vec{c}_1$ and $\vec{c}_2$ are two out-linked category name vectors of two linked articles. Apparently, the higher the out-linked category similarity of two articles the more relevant these two articles. As for three concepts mentioned above, the out-linked category similarity between "Data mining" and "Machine learning" is 0.656; the similarity between "Data mining" and "Database" is 0.213; and the similarity between "Machine learning" and "Database" is 0.157. It is very clear that "Data mining" is more related with "Machine learning" than with "Computer Network" and the relation between "Computer Network" and "Machine learning" is not as close as that between "Data mining" and "Machine learning", which accord with our human comprehension.

For each hyperlink in *Wikipedia*, we calculate the out-linked category similarity between two linked articles; and then, for each concept, we can rank all its out-linked concepts according to the out-linked category similarities of corresponding articles.

Distance based measure

The simplest distance based measure is the straightforward edge counting method, which measures semantic distance as the number of nodes in the taxonomy along the shortest path between two conceptual nodes [7]. So, with the acyclic graph formed by the *Wikipedia* hierarchical categorization structure, we define the category distance of two articles on this graph: suppose article A_1 belongs to category C_1 and article A_2 belongs to category C_2, the category distance of A_1 and A_2 is the shortest path of C_1 and C_2 on the categorization graph; and it is rational to say that the shorter the category distance the closer the relation of two articles. Accordingly, semantic relatedness is defined as the inverse score of the category distance. A normalized path-length measure taking into account the depth of the taxonomy in which the concepts are found is defined as:

$$Dis_{Category}(a_1, a_2) = \frac{length(c_1, c_2)}{D} \qquad (2)$$

Where $length(c_1, c_2)$ is the number of nodes along the shortest path between the two nodes (as given by the edge counting method), and D is the maximum depth of the taxonomy. Hereby, we could judge the pertinence of two articles by evaluate their category distance.

Linear combination of the three measures

After measuring each hyperlink in *Wikipedia* by above three methods, we can obtain three evaluation results of relatedness and then integrate them to get an overall relatedness evaluation result. We use linear combination of *TFIDF* similarity, out-linked category similarity and normalized category distance, which is:

$$S_{Overall} = \lambda_1 \cdot S_{TFIDF} + \lambda_2 \cdot S_{olc} + (1 - \lambda_1 - \lambda_2) \cdot Dis_{Category} \quad (3)$$

Where λ_1 and λ_2 are the weight parameters tuned from experiments. Later in Sec 5 we will explain how to adjust these weight parameters.

Then for each article in *Wikipedia*, we rank all its out-linked articles according to the overall relatedness evaluation of corresponding links. Thus, we get a relatedness ranking on out-linked concepts of each concept, and we deem the out-linked concepts with relatedness above certain threshold as associative ones for each concept.

Wikipedia, with its interwoven tapestry of articles, is a huge mine of information about concepts and hierarchical and associative relations: concepts represent the basic units of meaning; relations between them serve humans to organize and share their knowledge. Our thesaurus derived from *Wikipedia* maintains its coverage and accuracy. It has been successfully exploited for content-based categorization of large document collections, yielding a more perspicuous representation of text document.

4. Compiling Wikipedia Knowledge into the Text Document Representation

As we have mentioned before, the bag of words (*BOW*) approach only leverages the terms explicitly mentioned in text documents, thus fail to reflect relationships between important terms that do not co-occur literally. So integrating background knowledge to text documents may overcome the shortage of *BOW* approach. Moreover, *Wikipedia* is well-known for its most strength of containing much information about specific entities in the world, and such knowledge is not available through other electronic resources. Therefore, we build a general thesaurus from Wikipeida to exploit the background knowledge for text corpus. We first describe text document representation, then various strategies of background knowledge integration into the initial representation of text documents.

4.1. Text Document Representation

The text document representation is considered to be weighted bags of terms. Let $TF(d,t)$ be the absolute frequency of a term $t \in T$ in a document $d \in D$, where D is the set of documents and $T = \{t_1, \cdots, t_m\}$ is the set of all different terms occurring in D. First, stopwords are removed from T using a standard stopwords list[2], since stopwords are considered as non-descriptive terms within *BOW* approach. Then words in each document are stemmed using the Porter stemmer [8]. And the stemmed terms construct a vector representation $\vec{t}_d$ for each text document. The term vectors are denoted as $\vec{t}_d = (TF(d,t_1), \ldots, TF(d,t_m))$. After that, *TFIDF* (term frequency-inverted document frequency) weighs the frequency of each term in a document with a factor that discounts its importance when it appears in almost all documents. The *TFIDF* of a term t in document d is defined as:

$$TFIDF(d,t) = TF(d,t) * \log(|D| / DF(t)) \quad (4)$$

Where $DF(t)$ is the document frequency of term t that counts in how many documents where term t appears. After *TFIDF* weighting is applied, the term vector $\vec{t}_d = (TF(d,t_1), \ldots, TF(d,t_m))$ is replaced as $\vec{t}_d = (TFIDF(d,t_1), \ldots, TFIDF(d,t_m))$.

4.2. Text Document Enrichment

With the thesaurus built from *Wikipedia*, it is convenient to integrate background knowledge into text documents. As shown in Figure 1, firstly based on a filtered *Wikipedia* concept index, we search candidate concepts mentioned in each text document, and then add synonymies, hyponymies and associative concepts of these candidate concepts into documents. Thus, new concepts are added according to the content of original documents, and their purpose is to enrich the representation of original text documents. Then added concepts are new features for original documents which can be leveraged for categorization. Thus, related documents which do not share common terms literally may be enriched with the same concepts, as related documents connote the same background knowledge and concepts are add according to content of a text document. In the rest part of this section, we will introduce the steps in detail.

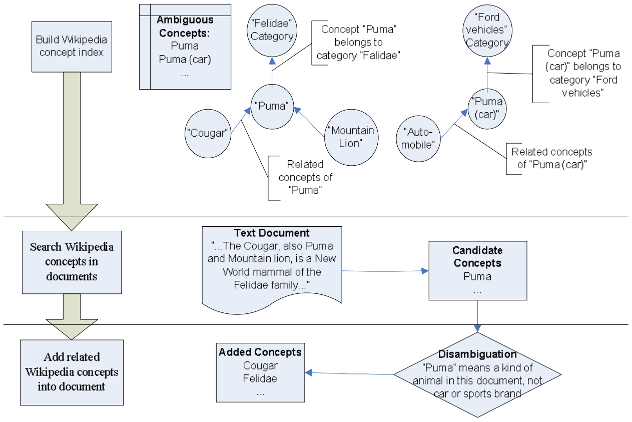
Build Wikipedia concept index
Ambiguous Concepts:
Puma
Puma (car)
...
"Felidae" Category
Concept "Puma" belongs to category "Falidae"
"Puma"
"Cougar"
"Mountain Lion"
Related concepts of "Puma"
"Ford vehicles" Category
Concept "Puma (car)" belongs to category "Ford vehicles"
"Puma (car)"
"Auto-mobile"
Related concepts of "Puma (car)"
Search Wikipedia concepts in documents
Text Document
"...The Cougar, also Puma and Mountain lion, is a New World mammal of the Felidae family..."
Candidate Concepts
Puma
...
Add related Wikipedia concepts into document
Added Concepts
Cougar
Felidae
...
Disambiguation
"Puma" means a kind of animal in this document, not car or sports brand

Figure 1: Document Enrichment Procedure

Coupled with the ability to enrich documents with concepts using thesauri, the approach addresses two main problems of natural language processing — synonymy and polysemy [2]. Enriching text documents with concepts from thesauri has two benefits. First it resolves synonyms, and second it introduces more general concepts which help identifying related topics. For instance, a document about "Puma" may not be related to another document about "Cougar" by classification algorithm if there are only "Puma" and "Cougar" in each term vector. But in our thesaurus, concept "Puma" and "Cougar" belong to category "Felines". So if the more general concept "Felines" is added to both documents, their semantic relationship is revealed.

4.2.1. Index Wikipedia concepts

[2] See http://www.aifb.uni-karlsruhe.de/WBS/aho/clustering

We define each title of *Wikipedia* articles as a *Wikipedia* concept, except some useless titles, such as "List of ISO standards", "1960s" and so on. So before indexing *Wikipedia* concepts, it is necessary to remove improper titles.

First, all titles of *Wikipedia* articles belonging to categories related to chronology, such as "Years", "Decades" and "Centuries" are removed. Second, because every title in *Wikipedia* must begin with a capital letter, we can judge whether a title is a concept by the following sequence of heuristic steps:

1. If the document title is a multiword title, check the capitalization of all words other than prepositions, determiners, conjunctions, or negations. If all the words are capitalized we considered it as a concept.

2. If the document title is one word title that occurs in its article more than 3 times, we consider it as a concept.

After removing improper titles, we build a concept index. Give a word, the index will find out all *Wikipedia* concepts containing this word.

Meanwhile, we gather all polysemous concepts, which are concepts holding multiple meanings, into the ambiguous concept set. For instance, the concept "Puma" is an ambiguous one, since it may refer to a kind of animal, car or something else.

4.2.2 Search Wikipedia concepts in documents

With the *Wikipedia* concept index, we search *Wikipedia* concepts mentioned in documents. The concepts mentioned in text documents are called candidate concepts. We search candidate concepts in documents as the following steps:

1. Split the document into vectors of term sequence by punctuations such as semicolon, interrogation, exclamatory point and full stop.

2. Find candidate concepts in each term sequence via window filtering condition described below.

3. Filter candidate concepts again to remove the concepts subsumed by other candidate concepts.

In each term sequence, the window filtering condition searches candidate concepts by Front Maximum Matching algorithm, and requires that every word of a concept much appear in the sequence within a window of certain length. For concepts of less than three words, the window length is the number of words in the concept; for concepts of more than three words, the window length is 1 plus the number of words in the concept.

$$LEN_{Window} = \begin{cases} WordCount(Concept), & WordCount(Concept) \le 2 \\ WordCount(Concept)+1, & WordCount(Concept) > 2 \end{cases} \quad (5)$$

Here is an example for the window filtering condition: the length of the filtering window for the concept "Ford Puma" is 2. As for the term sequences listed in Table 2, although all the four sequences mention "Ford Puma", only the first sequence will introduce the concept "Ford Puma" as a candidate one because it satisfies the window filtering condition: the two words "Ford" and "Puma" appear with the window of the length 2, and their order is consistent with their arrangement in the concept "Ford Puma"; whereas the concept "Ford Puma" won't be considered as a candidate one for the other three sequences.

Sent.1	The ***Ford Puma*** was a small coupe produced by the Ford Motor Company.
Sent. 2	The ***Ford*** *Racing* ***Puma*** was created in a limited run of just 500 by Tickford.
Sent. 3	Stylistically, the ***Puma*** followed ***Ford***'s New Edge design strategy
Sent. 4	The Puma was only sold in Europe and was supposedly replaced by the ***Ford*** StreetKa, which is based on the Fiesta just as the ***Puma*** was.

Table 2: Example for Window Filtering Condition

The window is to guarantee that all words of a concept must appear in a sequences within certain distance, which ensures the concept is truly mentioned in this document (consider the sentence "*Harrison* ***Ford****, a famous actor, was in a suit of* ***Puma*** *sportswear.*", although this sentence contains words "Ford" and "Puma", it doesn't talk about the car "Ford Puma").

4.2.3 Add Wikipedia concepts into document

After searching candidate concepts in documents, we add the related concepts of each candidate concept into documents. The related concepts of a candidate concept include its synonymies, hyponymies and associative concepts.

If a candidate concept belongs to the ambiguous concept set, that is to say the candidate concept is a polysemous one, it is necessary to do word sense disambiguation to find its most proper meaning mentioned in documents.

We adopt two strategies to do word sense disambiguation: the first one is to utilize text similarity for disambiguation; the second is to disambiguate with context.

Disambiguation with text similarity

This method is based on document similarity which is measured by term overlap to perform explicit word sense disambiguation. For instance, the Reuters document #15264 talks about copper mining, but the concept "Copper" in *Wikipedia* refers to several different meanings as listed in Table 3. The correct meaning of a polysemous concept may be found out by comparing the cosine similarity between *TFIDF* term vector of the text document and that of *Wikipedia* articles describing different meanings of the polysemous concept. As discussed in Sec. 3.1.4, the higher the cosine similarity of two *TFIDF* term vectors the more related these two text documents. Thus, for *Wikipedia* articles describing different meanings of a polysemous concept, the meaning described by the article with the highest *TFIDF* cosine similarity is considered as the most appropriate one. From Table 3, the *Wikipedia* article describing "Copper" has the max similarity with the Reuters document #15264, and then it is confirmed that the term "Copper" in document #15264 refers to the concept "Copper" in *Wikipedia*, not other concepts as "Copper (color)" or "Copper (I) oxide".

Meanings of "Copper"	**TFIDF Similarity with Reuters #15264**
Copper	0.339733115
Copper (color)	0.203722805
Copper (comic)	0.197735235
Copper(I) iodide	0.133150464
Copper(I) oxide	0.169211264
Copper(I)-thiophene-2-carboxylate	0.064311508

Copper(II) acetate	0.162892376
Copper(II) carbonate	0.20138639
Copper(II) fluoride	0.159417748
Copper(II) hydroxide	0.178744084
Copper(II) nitrate	0.158279119
Copper(II) oxide	0.208643492
Copper(II) sulfate	0.172548312

Table 3: The TFIDF similarity between Reuters document #15264 and the Wikipedia articles corresponding to different meanings of the term "Copper"

Disambiguation with context

Disambiguation with context is based on a method of conceptual distance like Agirre et al [9]. We give an example first, and then describe this method in detail.

Considering the sentence in Table 4, which is the correct meaning of the concept "Puma" referred in it, a kind of car or animal? In this sentence there also mentioned other *Wikipedia* concepts, such as "Cougar", "Mountain Lion" and "Felidae", in which "Cougar" and "Felidae" are also polysemous concepts. However, in *Wikipedia* there is a "Redirect" link from the concept "Mountain Lion" to the concept "Cougar". Then it's clear that the concept "Cougar" in this sentence refers to a kind of animal. And in *Wikipedia* the concepts "Cougar", "Puma" and "Felidae" belong to the same category "Felines". Therefore, it is sure that "Cougar", "Puma" and "Felidae" all refer to a kind of animal in this sentence. So the meaning of "Puma" is disambiguated by other concepts in the same context.

The cougar, also known as the ***puma*** *or mountain lion, is a New World mammal of the Felidae family.*

Table 4: Disambiguation with context

This disambiguation method is to discover relations between concepts within a context. And the relations between concepts are represented by the structural information of *Wikipedia*. Since our thesaurus has integrated the structural information of *Wikipedia* into itself, it is convenient to leverage the thesaurus for disambiguation. Here is the conceptual distance function between any two concepts in the thesaurus. For a concept C_1 and another concept C_2, their conceptual distance is defined as:

$$Dis_{Concept}(C_1, C_2) = \begin{cases} 1 & \text{If } C_1 \text{linksto} C_2 \\ Dis_{Category} & \text{Otherwise} \end{cases} \qquad (6)$$

Where "C_1 links to C_2" means C_1 is either a synonymy or an associate concept of C_2. In other words, if there is a link between C_1 and C_2, their conceptual distance is 1, otherwise is their category distance. In Table 4, the conceptual distance between "Cougar" and "Mountain Lion" is 1.

So if a sentence refers to a polysemous concept, first calculate the conceptual distance for each meaning of the concept with other non-polysemy concepts mentioned in this sentence; and then compute the average conceptual distance of each meaning, which is:

$$Dis_{Averaget}(C) = \frac{\sum_{j=0}^{n} Dis_{Concept}(C, C_j)}{n} \qquad (7)$$

Finally, the meaning with minimum average conceptual distance is considered as the most appropriate one.

However, a lot of sentences only mention at most one *Wikipedia* concept in them. So disambiguation with context is not always applicable for every polysemy. When disambiguation with context is not applicable, disambiguation with text similarity is adopted. If disambiguation with context is available, we take the average of two disambiguation results as combined result.

Here gives an example. The document #15264 from Reuters-21578 discusses a joint mining venture by a consortium of companies, and belongs to the category "copper". This document mentions several concepts as "copper", "mining" and "Teck Cominco" (which is a Canadian mining company). Table 5 shows the hyponymies, associative concepts and synonymies introduced into this document by these concepts.

Term	Hyponymies	Associative Concepts	Synonymies
mining	"mining companies", "mining companies"	"Open-pit mining", "Hard rock mining", "Sub-surface mining", "Surface mining"	"mine planning", "miner", "metal mining", "mine (industry)", "mineral engineering", "mineral extraction", "miners", "mining industry", "ore body"
copper	"chemical elements", "transition metals"	"Copper(II) carbonate", "Copper(II) oxide", "Copper extraction", "Native copper"	"copper (element)", "copper band", "copper sheet", "copper sheet metal", "cuprous", "cuprum", "copper mine", "cupper", "cupric"
Teck Cominco	"mining companies of canada", "s&p/tsx composite index"	"Mining", "Canadian Pacific Railway", "Kirkland Lake", "Ontario"	"Teck Cominco Ltd."

Table 5: The hyponymies, associative concepts and synonymies added into Reuters document #15264

When adding synonymies, associative concepts and hyponymies of a candidate concept into a text document, how many new concepts should be added? As for hyponymies, the direct hyponymies (which are the category names a concept directly belongs to) are most strongly related with a concept, as ancestor categories of the article corresponding to the concept are far too general; and the further the distance between ancestor categories and a *Wikipedia* concept the weaker their relations. For example, the concept "Puma" belongs to the category "Felines"; and the category "Felines" belongs to the category "Carnivores", which belongs to the category "Mammals". The relation between "Puma" and "Felines" are much stronger than those between "Puma" and "Carnivores" or "Mammals". Later, Sec 5.3 demonstrates the experiment results of adding different number of synonymies and hyponymies

5. Experiments

The evaluation was done with the *Wikipedia* snapshot dumped on November 30, 2006. After decompression, the resulting XML file was 8.6GB in size.

5.1. Processing Wikipedia data

As an open source project, the entire content of *Wikipedia* is easily obtainable. It is available in the form of database dumps that are released periodically, from several days to several weeks apart. The version used in this study was released on Nov. 30, 2006. The full content and revision history at this point occupy 70 GB of compressed data. We consider only the link structure and basic statistics for articles, which consume 1.9 GB (compressed).

We identified over four million distinct entities (articles and redirections) that constitute the vocabulary of thesauri. These were organized into 120,000 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.

Terms in Wikipedia	2250000
Concepts	1110111
Redirected Concepts	1020000
Categories	120000
Relations in Wikipedia	33060000
Redirect to Concept	1020000
Category to Subcategory	240000
Category to Concepts	3050000
Concept to Concept	28750000

Table 6: Content of Wikipedia

After filtered *Wikipedia* concepts as described in Sec 4.2, we got 627,255 concepts. Table 6 breaks down the data.

5.2. Data

Reuters-21578 [10]. Following common practice, we used the ModApte split (9603 training, 3299 testing documents) and two category sets, 10 largest categories and 90 categories with at least one training example and one testing example.

OHSUMED [11] is a subset of MEDLINE, which contains 348,566 medical documents. Each document contains a title, and about two-thirds (233,445) also contain an abstract. Each document is labeled with an average of 13 MeSH3 categories (out of total 14,000). Following Joachims [13], we used a subset of documents from 1991 that have abstracts, taking the first 10,000 documents for training and the next 10,000 for testing. To limit the number of categories for the experiments, we randomly generated 5 sets of 10 categories each.

20 Newsgroups (20NG) [12] is a well-balanced dataset of 20 categories containing 1000 documents each.

5.3. Experiment Results

The linear form of Support Vector Machine (SVM) [14] classification model is used to learn model to classify documents. We measured text categorization performance using the precision-recall break-even point (BEP). For the Reuters and OHSUMED datasets, we report both micro-averaged and macro-averaged BEP, since their categories differ in size substantially (micro-averaged BEP operates at the document level and is primarily affected by categorization performance on larger categories; whereas macro-averaged BEP averages results over categories, and thus small categories have large impact on the overall performance). Following established practice, we used a fixed data split for the Reuters and OHSUMED datasets, and consequently used macro sign test (S-test) [15] to assess the statistical significance of differences in classifier performance. For 20NG dataset, we performed 4-fold cross-validation, and used paired t-test to assess the significance.

5.3.1 Parameter Tuning

As mentioned in Sec. 3.1, we adopt three kinds of methods to measure the relatedness between *Wikipedia* concepts, and use linear combination of *TFIDF* similarity, out-linked category similarity and normalized category distance to merge the results of three methods. Here we introduce how to tune λ_1 and λ_2 of Equation 3.

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. No one among the three assessors could access the labeling results of others, who are graduate students and have good command of the English language. After labeling, each out-linked concept in the 10 articles is labeled with 3 relevance tags, and we use the average value as the final relatedness value. For example, if one user labels two linked concepts as neutral and the other two user label them as relevant, then the final relatedness of the pair of linked concepts is 1.67 ((1+2+2)/3). Based on the labeled tuning data, we calculate *TFIDF* similarity, out-linked category similarity and normalized category distance between the 10 concepts and the concepts in these concept documents. We tune the value of λ_1 and λ_2 from 0.1, 0.2, up to 1.0, and thus we can find the proper values of λ_1 and λ_2, with which result of linear combination matches user evaluation result best. From experiments, λ_1 is set to 0.4 and λ_2 is set to 0.5.

5.3.2 The Effect of Document Enrichment

As described in Sec. 4.2, when enriching documents, we first find candidate concepts that mentioned in a text document, and then enrich documents of new concepts introduced by candidate concepts. We have considered different strategies: adding synonymies, adding hyponymies and adding associative concepts. And how many new concepts should be added? Here demonstrated the effect of classification with documents of adding different kinds of concepts and different number of concepts.

Table 7 demonstrates the performance of document augment with hyponymies. We first add the direct hyponymies (which are category names a candidate concept directly belongs to) for each candidate concepts, and then hyponymies of both first and second level (which are parent category names of the direct category a candidate concept belongs to), until hyponymies within 5 levels. In Table 7, “Baseline” means we don’t add any concepts into documents; and “H1” means adding direct hyponymies into documents; and “H2” means adding hyponymies of both first and second level, so does for “H3” to “H5”. Then we found that adding direct hyponymies and hyponymies within first two levels achieves the best result on categorization, and adding more hyponymies of further levels even deteriorates the classification result.

Dataset	Reuters		20NG		Ohsumed	
	Micro	Macro	Micro	Macro	Micro	Macro
Baseline	0.877	0.605	0.868	0.865	0.602	0.548
H1	0.891	0.623	0.904	0.892	0.658	0.585
H2	0.883	0.619	0.898	0.886	0.642	0.574
H3	0.878	0.607	0.881	0.879	0.631	0.568
H4	0.871	0.601	0.875	0.868	0.617	0.553
H5	0.868	0.593	0.869	0.857	0.604	0.540

Table 7: The effect of adding hyponymies

Table 8 shows the result of enriching documents with associative concepts. For each candidate concepts, we append documents of its top 5, 10, 15, 20 and 25 most similar associative concepts. “Baseline” still means we don’t add any concepts into documents; and “A5” means adding 5 most associative concepts into documents, so does for “A10” to “A25”. Likewise, we found that adding 5 or 10 most associative concepts brings best classification result, whereas adding more associative concepts even worse than the baseline. And overall, the result of adding associate concepts is better than that of adding hyponymies.

Dataset	Reuters		20NG		Ohsumed	
	Micro	Macro	Micro	Macro	Micro	Macro
Baseline	0.877	0.605	0.868	0.865	0.602	0.548
A5	0.907	0.629	0.915	0.896	0.667	0.590
A10	0.899	0.621	0.908	0.887	0.656	0.578
A15	0.884	0.617	0.896	0.875	0.639	0.561
A20	0.879	0.608	0.889	0.868	0.628	0.553
A25	0.871	0.599	0.878	0.859	0.611	0.542

Table 8: The effect of adding associative concepts

For adding synonymies, we found that this method fails to bring as prominent improvement as former two strategies, as showed in Table 9. Since we can’t rank synonymies of a given candidate concept, we just add all its synonymies into documents, which inevitablely brings some noise into documents. As mentioned in Sec. 3.1.1, “Redirect” link also copes with capitalization and spelling variations, abbreviations, synonyms, colloquialisms, and scientific terms. For instance, the document #15264 from Reuters-21578 is talking about copper mining, and the synonymies of the word “copper” in this document are “copper (element)”, “copper band”, “copper sheet”, “copper sheet metal”, “cuprous”, “cuprum”, “copper mine”, “cupper”, “cupric” and “element 29”. We found that “cupper” maybe a misprint of “copper”, and “cupper” should not be added.

Dataset	Reuters		20NG		Ohsumed	
	Micro	Macro	Micro	Macro	Micro	Macro
Baseline	0.877	0.605	0.868	0.865	0.602	0.548
Add Synonymies	0.854	0.597	0.852	0.858	0.524	0.515

Table 9: The effect of adding synonymies

Finally we tried add both hyponymies and associative concepts together into documents and found out that, when adding into documents direct hyponymies and 5 most associative concepts for each candidate concept, this strategy achieves more improvement on categorization, as showed in Table 10.

Dataset	Reuters		20NG		Ohsumed	
	Micro	Macro	Micro	Macro	Micro	Macro
Baseline	0.877	0.605	0.868	0.865	0.602	0.548
Combined	0.912	0.631	0.917	0.904	0.672	0.593

Table 10: The effect of adding both hyponymies and associative concepts

6. Conclusion and Future Works

In this paper, we first propose a new way to build thesaurus from *Wikipedia*, and utilize the thesaurus to facilitate text categorization. *Wikipedia* is a huge resource of encyclopedia knowledge, and we mine relation of concepts from it to build our thesaurus. Then, we enrich documents with the related concepts of candidate concepts mentioned in documents. And when enriching documents, we perform explicit disambiguation to find out the correct meaning of each polysemous concept expressed in documents. By doing so, background knowledge can be introduced into documents, which remedies the shortage of *BOW* approach. And experiments demonstrate that our method brings a lot improvement.

For future works, we should first try to meliorate the effect of adding sysnonymies by filtering “Redirect” links. After removing useless redirect links such as spelling variations and keeping significative ones as sysnonyms and abbreviations, we think adding sysnonymies into text documents won’t bring as much noise as before, and its effect will be better.

The disambiguation strategies we adopt can be further improved. Since the thesaurus has build a relation graph for each concept, which includes its synonymies, hyponymies and associative concepts, when disambiguation, the graph can be utilized.

Further out, *Wikipedia* contains so much information, and our thesaurus only explores part of its resources. Other information in *Wikipedia* can be minded. For example, the link relation in *Wikipedia* is such a meaningful resource, and our thesaurus hasn’t taken advantage of anchor text. Since anchor texts of links are also synonymies of the titles of linked articles, our thesaurus can be expanded. Moreover, *Wikipedia* includes articles of many languages, so cross language information retrieval can be fulfilled.

In a word, *Wikipedia* is such an abundant resource, it has outweighed many traditional resources, and its exploitation is booming.

7. Reference

[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. In 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. 2006. Using encyclopedic knowledge for named entity disambiguation. In EACL-06.

[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. 1997. 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. ECML'98, pp. 137–142.

[14] F. Sebastiani. 2002. Machine learning in automated text categorization. ACM Computing Surveys 34(1). 2002. pp.1–47.

[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. John Benjamins. 2000.

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