Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge

Xia Hu [†,‡], Nan Sun [‡], Chao Zhang[‡], Tat-Seng Chua[‡]

[†] School of Computer Science and Engineering
Beihang University
Beijing 100191, P.R. China

[‡] School of Computing
National University of Singapore
Computing 1, Singapore 117590

{huxia, sunn, zhangcha, chuats}@comp.nus.edu.sg

ABSTRACT

Clustering of short texts, such as snippets, presents great challenges in existing aggregated search techniques due to the problem of data sparseness and the complex semantics of natural language. As short texts do not provide sufficient term co-occurrence information, traditional text representation methods, such as "bag of words" model, have several limitations when directly applied to short text tasks. In this paper, we propose a novel framework to improve the performance of short text clustering by exploiting the internal semantics from the original text and external concepts from world knowledge. The proposed method employs a hierarchical three-level structure to tackle the data sparsity problem of original short texts and reconstruct the corresponding feature space with the integration of multiple semantic knowledge bases – Wikipedia and WordNet. Empirical evaluation with Reuters and real web dataset demonstrates that our approach is able to achieve significant improvement as compared to the state-of-the-art methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*

General Terms

Algorithms, Experimentation

Keywords

Clustering, Short Texts, Syntactic Structure, Semantic Knowledge Bases

1. INTRODUCTION

Aggregated search aims to gather the search results from various resources and present them in a succinct format to improve the robustness and usability of the systems. One of the key issues in state-of-the-art aggregated technology is "How should the information be presented to the user?"[1]. Traditionally, browsing through search results in the form of a ranked list is inconvenient for users to effectively locate their interests. To address this problem, many research and commercial aggregated search systems, such as DIGEST [26] and Clusty[2], provide clustering of relevant search results to make the information more systematic and manageable. Consequently, these systems facilitate users' quick grasping of their interests by examining the overview of the subtopics provided by the clustering module.

Short texts, such as the snippets, product descriptions, QA passages and image captions etc., have played important roles in current Web and IR applications. Successful processing short texts is essential for aggregated search systems. However, unlike standard texts with lots of words and their statistics, short texts, which only consist of a few phrases or 2–3 sentences, especially present great challenges in clustering. They do not provide sufficient word co-occurrence or context shared information for effective similarity measure [23], which is the basis of clustering methods [15]. Therefore, the conventional texts clustering methods may fail to achieve satisfactory results when they are directly applied to short text tasks [21].

To tackle the data sparseness problem, several methods have been proposed in the literatures. One is the basic representation of texts, called surface representation [18, 19], exploits phrases in the original texts from different aspects to preserve the contextual information. However, NLP techniques, such as parsing, are not employed as it is time consuming to apply such techniques to analyze the structure of standard text in detail. As a result, the methods fail to perform deep understanding of the original text. Another limitation of such methods is that they did not use world knowledge, which has been found to be useful in dealing with the semantic gap in text representation [9]. For example, the first snippet returned by Google using "Sun" as the query does not contain any words or phrases related to "Oracle ", while we learn that these two companies are highly relevant from Sun's homepage via the link of the snippet. Because they have no common words or phrases, this snippet can not be successfully clustered into the "Oracle" related clusters. Thus, one obvious approach is to enrich the contexts of basic text segments by exploiting world resources and such methods have been found to be effective in narrowing the semantic gap in different tasks. Urena et al. [28] showed that the integration of WordNet effectively improved the perfor-

CIKM'09, November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

[1] http://www.yr-bcn.es/sigir08

[2] Previously known as Vivisimo. http://www.clusty.com/

mance of text classification and word sense disambiguation tasks. Gabrilovich and Markovitch [11] proposed to compute text semantic relatedness by representing the meaning of text as a weighted vector of Wikipedia-based concepts.

In this paper, we present a novel framework to improve the clustering of short texts by incorporating both the rich internal and external semantics. Internal semantics aim to provide a deep understanding of the original short texts and external semantics incorporate the concepts derived from the world knowledge to reduce the semantic gap. Our framework consists of three steps. First, we obtain the internal semantics based on hierarchical representation of the original short text by applying NLP techniques. *Original features* and *seed phrases* are carefully extracted from different levels of the hierarchical structure. *Original features* serve as part of the feature space for clustering and *seed phrases* provide a solid basis for feature generation in next step. Second, a phrase selection approach is introduced to eliminate the information duplicate among *seed phrases* by measuring the semantic similarity between them. Based on the *seed phrases*, we employ a feature generation strategy that leverages multiple resources and utilizes the advantages of each knowledge base, i.e. Wikipedia and WordNet, to generate high quality external semantics. These external semantics serve as the *external features*, in conjunction with the *original features* generated from the first step to construct the feature space for clustering. Finally, a feature selection method is proposed to avoid negative impact of the huge number of features from world knowledge. In our method, based on the hierarchical structure, we elaborately separate the phrases in the original short texts from different granularity to construct the *original features* and *seed phrases*. The latter are used as the basis for generating the *external features* implied in the world resources. In this way, both the internal and external information are better utilized based on the hierarchical structure in the proposed framework, and their contribution are extensively exerted. The empirical results on two datasets using different clustering algorithms show the effectiveness of our proposed method.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 presents the architecture of our proposed framework. Details of the proposed approaches are given in Sections 4, 5 and 6. Experimental results are presented in Section 7. Section 8 concludes the paper with directions for future work.

2. RELATED WORK

Many methods have been proposed to improve the representation of standard text for clustering and classification. These methods can generally be divided into two categories. One category is the traditional text representation methods called surface representation. Terms [19], name entities [18] and phrases [3] were extracted from the original text to construct the feature space. Another category is to enrich the text representation based on "bag of words" model by generating external features from linguistic and collaborative knowledge bases. Hotho et al. [14] observed that additional features from WordNet can improve clustering results. Gabrilovich and Markovitch [9, 10] analyzed the documents and mapped them onto the ontology concepts of Wikipedia and ODP ("Open Directory Project"), which in turn induced a set of features that augment the standard "bag of words". The experimental results of integrating collaborative knowledge bases show improvements as compared to the "bag of words" baselines in different tasks.

However, these approaches have several inherent limitations. The surface representation based techniques encounter the common semantic gap problem due to the lack of world knowledge [9]. As background knowledge, Dave et al. [7] utilized WordNet synsets independently as additional features for document representation and found that the performance of clustering decreased in his experiments; while the use of collaborative knowledge bases by current systems are limited to the user-defined categories and concepts in those repositories [15].

Several clustering techniques were employed to place the search engine snippets to their highly relevant topic-coherent groups. Some of the methods [5, 13] first clustered the snippets and then summarized each cluster to generate a cluster label. In contrast, other methods [30, 31] first extracted some common phrases from the set of snipeets as the cluster label and then created clusters according to these key phrases. However, these two kinds of methods highly rely on the common key phrases appearing in the texts and ignore the implicit semantic relationship between the phrases.

World knowledge bases have been found useful in improving the short text representation. Kohomban and Lee [17] built a word sense disambiguation system to tackle the data scarcity problem. This system trains the classifier using grouped senses for verbs and nouns according to the top-level synsets from WordNet and is able to effectively pool the training cases across senses within the same synset. Sahami et al. [25] addressed the data sparseness by leveraging web search results to provide greater context for short texts. This method shows the effectiveness for suggesting related queries to search engine users in a large-scale system. Recently, some methods were proposed to tackle the problems of data sparseness and semantic gap in short texts clustering or classification by exploiting world knowledge. Somnath et al. [1] proposed a method to enrich short texts representation with additional features from Wikipedia. Although this method only used the titles of Wikipedia articles as additional external features, it showed improvement in the accuracy of short texts clustering . Phan et al. [23] presented a framework for building classifiers that deal with short texts from Web and achieved significant quality enhancement. The underlying idea of the framework is to collect a large-scale external data collection, and then build a classifier on both labeled data and external semantics for each classification task.

3. THE GENERAL FRAMEWORK

In this Section we introduce the proposed framework that aims to improve the clustering of short texts by exploiting the internal and external semantics. The workflow consists of three consecutive phases, including Hierarchical Resolution, Feature Generation and Feature Selection, as shown in Figure 1. In our framework, internal semantics represent the features from the original text by employing the three-level hierarchical structure, while external semantics represent the features derived from external knowledge bases.

For ease of illustration, in Figure 1, we present an example for reconstructing feature space of a Google snippet, which describes a famous movie "The Dark Knight".

"Jul 18, 2008 ... It is the best American film of

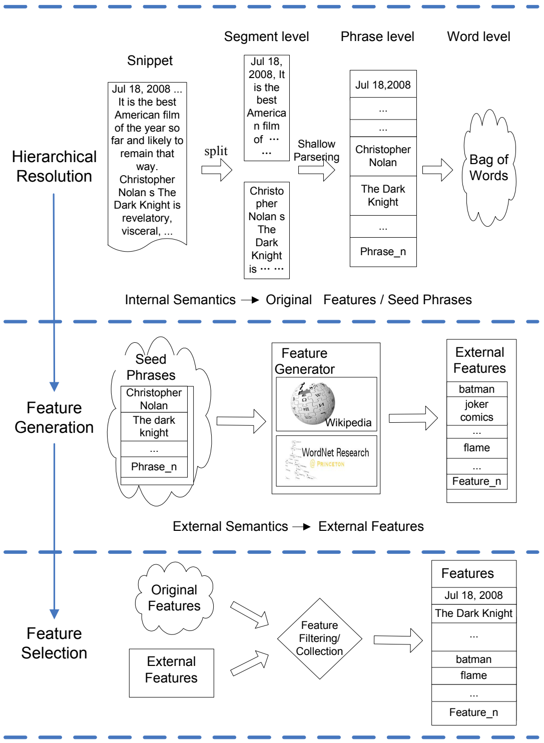

Figure 1: Framework for feature constructor

> the year so far and likely to remain that way. Christopher Nolan's The Dark Knight is revelatory, visceral ..."

Hierarchical Resolution Phase Short texts have the characteristics of sparsity, noisy and non topic-focus due to their limited length. When using "bag of words" model to represent short texts, it neglects contextual information and hence often leads to synonymy and polysemy problems [15]. In another way, the "bag of phrases" model in representation of short text is insufficient to provide enough term occurring information for clustering [16]. Thus a method which can better make use of the limited original text is necessary for such tasks. Many NLP techniques have achieved great success by using parser to analyze a sentence [4]. Therefore, inspired by the structure of parsing tree, we resolve the short text in a hierarchical view to extract the three-level internal semantics by employing NLP techniques. Each level of internal semantic features have their different characteristics, and together, they preserve the syntactic structure of short text from multiple aspects. From the hierarchical structure, we extract the *original features* which serve as part of the feature space for clustering, and *seed phrases*, which provide an informative basis for generating external semantics in the next phase.

Feature Generation Phase Internal semantics are extracted from the original texts in hierarchical resolution phase, however, they are still inadequate for the representation of short texts due to the semantic gap. It is difficult to determine whether two texts are semantically similar only by considering their original term co-occurring information. Therefore, we propose to employ feature generation techniques to enrich their representation space by leveraging repositories of world knowledge.

Feature generation consists of two steps, the construction of basic features and the generation of *external features*. The basic features should be informative for generating diverse external semantics and effective in avoiding producing noisy or redundancy features. Therefore, we first employ a similarity measure algorithm to eliminate duplicates in the *seed phrases* and use the remaining ones as the basic features for feature generation. For each *seed phrase*, we employ rules to measure whether WordNet or Wikipedia is more appropriate as semantic knowledge base. For Wikipedia pages, we not only use the explicit concepts, such as the titles and links terms, but also extract the hidden topics [23] to be incorporated into the external semantics. Also, the lexical features generated by WordNet are added as a complement.

Feature Selection Phase Features generated from world knowledge are unstructured and the huge number of *external features* leads to the "curse of dimensionality", which brings in negative impact to the feature space for clustering. Therefore, feature selection step is employed to refine the unstructured features derived from Wikipedia and to ensure that the reconstructed feature space is compact and effective for clustering.

In next Sections, we will present the details of these three phases.

4. HIERARCHICAL RESOLUTION

Original text normally contains precise and valuable information. The snippet in Figure 1 contains information on the description ("the best American film"), director ("Christopher Nolan") and title ("The Dark Knight") of the film. However, the "bag of words" approach, which ignores the contextual information of the text, is not able to capture the rich semantics of this short text. On the other hand, using text segments generated by spliter or chunker to represent the text is too sparse to locate the centroids of information from the noisy text [21]. Therefore, we propose to exploit the internal semantics of short texts, which not only preserves the contextual information but also avoids data sparsity.

4.1 Hierarchical Feature Extraction

A snippet typically comprises two or three sentences. In NLP tasks, people usually employ parsing to mine the syntactic structure contained in the sentences [4]. Figure 2 illustrates an example of the syntax tree[3] for the snippet in Figure 1. From this syntactic structure, we can see the snippet contains three important components, including Syntax Node(S, sentence), Branch(VP, NP verb phrase & noun phrase) and Leaf(words) of the tree. To preserve the original information conveyed by the snippet, we should make full use of these three components from the syntax tree. Under this scenario, we propose a method to decompose the original text into a three-level top-down hierarchical structure — *segment level*, *phrase level* and *word level*.

Segment Level: From the observation of short texts in Web applications, they do not always comprise fully structured sentences but segments like "Jul 18, 2008" as shown in Figure 2. We broadly define a text segment or sentence

[3] Due to limitation of space, we present only part of the syntax tree in Figure 2.

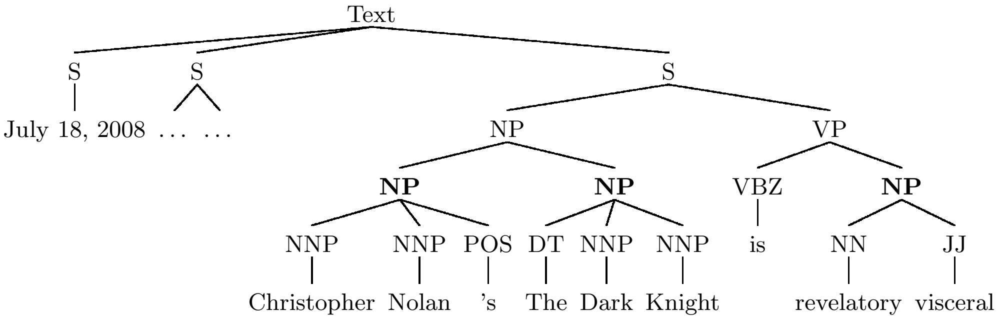

Figure 2: Syntax tree of the snippet in Figure 1

as a single unit in this level. The text is split into segments with the help of punctuations. Each segment contributes to provide a subtopic or one aspect of the original text.

Segment level features are informative and have been found to be beneficial in generating quality *external features* [1]. However, the features at this level are generally ambiguous and often fail to convey the exact information to represent the short text [33]. Therefore, we need to further exploit the internal semantics contained in the original text from the other two components of the syntax tree.

Phrase Level: When people speed-read through a text, they do not fully parse the sentence but instead look for "key phrases" [20]. Thus, shallow parsing [12] is adopted to divide sentences into a series of words that together compose a grammatical unit, mostly NP(noun phrase), VP(verb phrase) and PP(preposition phrase). The output of shallow parsing to the snippet in Figure 1 is as below:

- *Sentence1* : [NP July 18 2008]
- *Sentence2* : [NP It] [VP is] [NP the best American film] [PP of] [NP the year] [ADVP so far] and/CC [ADJP likely] [VP to remain] [NP that way]
- *Sentence3* : [NP Christopher Nolan 's] [NP The Dark Knight] [VP is] [NP revelatory visceral]

After stemming and removal of stop-words from the phrases generated by the shallow parser, the NP and VP chunks are employed as *phrase level* features.

Phrase level features are informative and more focused on one concept. They can readily be mapped to the relevant articles in world knowledge bases. Inevitably, there will be many information duplicates between *segment level* and *phrase level* features. We rely on the *seed phrase* selection step to eliminate the redundant features, preserving only those that genuinely characterize the text. Employing features at this level independently to represent text is too sparse due to the limited length of short text. Therefore, we incorporate *word level* features to better represent the short text.

Word Level: The cost of full parsing to analyze the sentence is expensive in both time and resources. Therefore, we do not use traditional syntactic technologies to obtain key words but decompose the *phrase level* features directly. We choose the non-stop words contained in NP and VP chunks to build the *word level* features.

Features at this level further remove the meaningless words in the short texts, thus offering a more effective feature space than "bag of words" model. They serve as additional features to represent the text and tackle the data sparseness to some degree. The negative characteristic of *word level* features is that they are too general to generate meaningful *external features.*

4.2 Original Feature Extraction

The three-level features from different granularities not only preserves the word ordering information in the short text, but also avoids data sparseness problem to some extent. As mentioned before, the *segment level* features are not appropriate to represent the text due to their sparseness. Thus, we extract features at *phrase level* and *word level* to compose our *original feature* set. First, the *phrase level* features, which contain the original contextual information implied in the hierarchies, are exploited to tackle one of the main impediments in NLP — polysemy. For example, the word "knight" may be regarded as "man to whom the sovereign has given a rank of honour" or "man raised to honourable military rank". The phrase "The Dark Knight" builds relationship with the movie and clearly indicates that the word "knight" here refers to the second meaning. Another example is that although every single word in "July 18 2008" is general and meaningless, when we consider this phrase as a unit, it is possible to build connection with topics of "movie" and "the dark knight" from some top ranking results returned by Google. Second, the *word level* features contribute as a complement to avoid the problem of data sparseness. The features from two levels of the hierarchical structure support each other to comprise the *original feature* space.

5. FEATURE GENERATION

Consider again the snippet in Figure 1, even by mining the original text, it is still inadequate to build the semantic relationship with other relevant concepts. For example, "The Dark Knight" and "batman" are different names of one movie, but they cannot be linked as the same concept without additional information from external knowledge. To narrow the semantic gap, we propose to mine external features to enrich the text representation.

In this Section, we present two steps of feature generation: the extraction of *seed phrases* from the internal semantics and the generation of *external features* from *seed phrases.*

5.1 Seed Phrase Selection

Among internal semantics, features at *segment level* and *phrase level* are informative to cover the key subtopics described in the short texts. We thus use features at these two levels to construct the *seed phrases*. However, there are redundancies between these two kinds of features as *phrase level* features are in a way derived from *segment level* features. For example, the *segment level* feature "Christopher Nolan's The Dark Knight is revelatory visceral" generates three *phrase level* features [NP Christopher Nolan's], [NP The Dark Knight] and [NP revelatory visceral]. If we employ all these features as *seed phrases*, they would produce many duplicate information between the *segment level* feature and [NP The Dark Knight]. Therefore, we propose to measure the semantic similarity between *phrase level* features and its parent *segment level* feature to eliminate information redundancy.

Several methods have been proposed to calculate the semantic similarity between words [27] or associations [2] using web search. However, along with the increasing scale of the web, the page counts provided by some commercial search engines are not so reliable [6]. Thus instead of simply using the search engine page counts, we propose a phrase-phrase semantic similarity measure algorithm using co-occurrence double check in Wikipedia to reduce the semantic duplicates. For Wikipedia we download the XML corpus [8], remove xml tags and create a Solr [4] index of all XML articles.

Let P denotes a *segment level* feature, $P = \{p_1, p_2, \ldots, p_n\}$, where p_i denotes the *phrase level* feature contained in P. The *segment level* feature is too sparse to calculate its frequency directly. Therefore, we calculate the semantic similarity between p_i and $\{p_1, p_2, \ldots, p_n\}$ as $InfoScore(p_i)$ instead. The p^* which has the largest similarity with other features in P will be removed as the redundant feature.

Given two phrases p_i and p_j, we use p_i and p_j separately as query to retrieve top C Wikipedia pages from the built index. The total occurrences of p_i in the top C Wikipedia pages retrieved by query p_j is denoted as $f(p_i|p_j)$; and we define $f(p_j|p_i)$ in a similar manner. The total occurrences of p_i in the top C Wikipedia pages retrieved by query p_i is denoted as $f(p_i)$, and similarly for $f(p_j)$. The variants of three popular co-occurrence measures [6] are defined as below:

$$WikiDice(p_i, p_j) = \begin{cases} 0 & \text{if } f(p_i \mid p_j) = 0 \\ & \text{or } f(p_j \mid p_i) = 0 \\ \frac{f(p_i|p_j)+f(p_j|p_i)}{f(p_i)+f(p_j)} & \text{otherwise} \end{cases}, \quad (1)$$

where WikiDice is a variant of the Dice coefficient.

$$WikiJaccard(p_i, p_j) = \frac{min(f(p_i \mid p_j), f(p_j \mid p_i))}{f(p_i) + f(p_j) - max(f(p_i \mid p_j), f(p_j \mid p_i))}, \quad (2)$$

where WikiJaccard is a variant of the Jaccard coefficient.

$$WikiOverlap(p_i, p_j) = \frac{min(f(p_i \mid p_j), f(p_j \mid p_i))}{min(f(p_i), f(p_j))}, \quad (3)$$

where WikiOverlap is a variant of the Overlap(Simpson) coefficient.

For ease of comparison, all the $\frac{n^2}{2}$ WikiDice similarity scores are normalized into values in $[0, 1]$ range using the linear normalization formula defined below:

$$WD_{ij} = \frac{WikiDice_{ij} - min(WikiDice_k)}{max(WikiDice_k) - min(WikiDice_k)}, \quad (4)$$

where k is from 1 to $\frac{n^2}{2}$. We again define WJ_{ij} and WO_{ij} in a similar manner for WikiJaccard and WikiOverlap respectively. A linear combination is then used to incorporate the three similarity measures into an overall semantic similarity between two phrases p_i and p_j, as follows:

$$WikiSem(p_i, p_j) = (1 - \alpha - \beta)WD_{ij} + \alpha WJ_{ij} + \beta WO_{ij}, \quad (5)$$

where α and β weight the importance of the three similarity measures. As text clustering is an unsupervised method, where we do not have any labeled data to tune the parameters. We thus empirically set α and β to equal weight.

For each *segment level* feature, we rank the information score defined in Equation 6 for its child node features at *phrase level*.

$$InfoScore(p_i) = \sum_{j=1, j \neq i}^{n} WikiSem(p_i, p_j). \quad (6)$$

Finally, we remove the *phrase level* feature p^*, which delegates the most information duplicate to the *segment level* feature P, and it is defined as:

$$p^* = \arg \max_{p_i \in \{p_1, p_2, \ldots, p_n\}} InfoScore(p_i). \quad (7)$$

5.2 Background Knowledge Bases

Wikipedia, as background knowledge, has a wider knowledge coverage than WordNet and is regularly updated to reflect recent events. On the other hand, as the construction of WordNet follows theoretical model or corpus evidence, it contains rich lexical semantic knowledge [32]. Under this scenario, we take Wikipedia as the principle semantic knowledge source and WordNet as the secondary one.

Gabrilovich and Markovitch [10], as well as Hu et al. [15] preprocessed the Wikipedia data to collect Wikipedia concepts. Preprocessing of Wikipedia ignores the valuable contextual information of Wikipedia plain texts and always encounters problems when mapping the original text to appropriate concepts. Therefore, in this study we preserve the original pages of Wikipedia with the built-in Solr index as described in Section 5.1.

5.3 Feature Generator

The *external feature* generation algorithm is illustrated in Figure 3. Given a *seed phrase*, we first employ heuristic rules to distinguish which knowledge base is more appropriate. If the phrase contains more than one non-stopword, then this phrase is regarded as containing enough information to reflect one aspect or subtopic of the short text. For this kind of phrase, we can retrieve accurate Wikipedia pages with the help of Solr search engine. On the other hand, phrase that has one or zero non-stopword is regarded as too general to generate accurate concepts from Wikipedia. We thus use WordNet as complement to deal with such phrases from lexical aspect.

In order to retrieve the appropriate pages from the large scale Wikipedia corpus, we derive queries based on *seed*

[4] http://lucene.apache.org/solr/

```
Algorithm 1: GenerateFeatures($S$)
input  : a set $S$ of seed phrases
output: external features $EF$
EF ← null
for seed phrase $s \in S$ do
    if s.non-stop >1 then
        if $s \in$ Segment level then
            $s$.Query ← SolrSyntax($s$, OR)
        else
            $s$.Query ← SolrSyntax($s$, AND)
        WikiPages ← Retrive($s$.Query)
        $EF \leftarrow EF$ + Analyze(WikiPages)
    else
        $EF \leftarrow EF$ + WordNet.Synsets($s$)
return $EF$
```

Figure 3: *External feature* generation scheme

phrase arising from *segment level* or *phrase level* separately. As the key information of *seed phrases* from *phrase level* is more focused, we build the "AND" query which requires the retrieved pages to contain every term in the phrase. On the other hand, the *seed phrases* from *segment level* are informative but sparse, we thus build "OR" query[5] which means there is no guarantee that the retrieved Wikipedia pages will contain every term in the phrase. We use these two kinds of queries to retrieve the top ω articles from the Wikipedia corpora. Similar to previous work [1], we extract titles and bold terms (links) from the retrieved Wikipedia pages to serve as part of the *external features.* To discover the intrinsic concepts hidden in the plain texts, we adopt an effective key phrase extraction algorithm — Lingo [22], which uses algebraic transformations of the term-document matrix and implements frequent phrase extraction using suffix arrays. The key phrases extracted from the Wikipedia pages are added to the *external feature* space. For example, we may obtain extrinsic concepts "batman" for the phrase "The Dark Knight" and the intrinsic concepts like "fireworks" or "joker" by mining the related pages.

If the phrase contains only one non-stopword (e.g. "in his car"), WordNet synsets are applied to extract similar concepts (e.g. "atuo", "automobile", "autocar") of the substantive ("car").

With this scheme, we can easily tackle the phrase sense synonymy from both the semantic and lexical aspects. The semantic synonymy of phrases is handled by the *external features* generated from Wikipedia, while the similar "neighbors" from WordNet synsets help to tackle the lexical synonym. For example, the phrase "batman" is generated by "The Dark Knight" and they are highly semantically related; and we can also distinguish that "in his car" talks about the same scene as "in his automobile" with the help of the WordNet.

6. FEATURE SELECTION

As the construction of Wikipedia follows the non-binding guidelines and the data quality is only under social control by the community [32], it often leads to noise in the corpus. Meanwhile, a single text may generate a huge number of features. These overzealous *external features* bring adverse impact on the effectiveness and dilute the influence of valuable original information. Therefore, we conduct feature filtering to refine the unstructured or meaningless features and apply feature collection to avoid aggravating the "curse of dimensionality".

Feature Filtering: We formulate empirical rules to refine the unstructured features obtained from Wikipedia pages, some typical rules are as follows:

- Remove features generated from too general *seed phrase* that returns a large number (more than 10,000) of articles from the index corpus.
- Transform features used for Wikipedia management or administration, e.g. "List of hotels"→"hotels", "List of twins"→"twins".
- Apply phrase sense stemming using Porter stemmer [24], e.g. "fictional books"→"fiction book".
- Remove features related to chronology, e.g. "year", "decade" and "centuries".

External Feature Collection: We have obtained n_1 *original features* in Section 4.2, and now collect n_2 *external features* to construct $n_1 + n_2$ dimension feature space for clustering. The number of *external features* we need to collect is determined by:

$$n_2 = \frac{n_1 \times \theta}{1 - \theta}, \tag{8}$$

where θ is the fraction of *external features* to the whole feature space in an interval $[0, 1]$. In the extreme cases, $\theta = 0$ means the feature space contains no *external features* and $\theta = 1$ indicates the features in the feature space are all from *external features.*

First, *tf-idf* weights of all the generated features are calculated. One *seed phrase* $s_i (0 < i \le m)$ may generate k *external features* $\{f_{i1}, f_{i2}, \ldots, f_{ik}\}$. In order to explore the diversity of the *external features*, we select one feature for each *seed phrase.* Thus m features are collected as follows:

$$f_i^* = \arg \max_{f_{ij} \in \{p_{i1}, p_{i2}, \ldots, p_{ik}\}} tf\_idf(f_{ij}). \tag{9}$$

Second, the top $n_2 - m$ features are extracted from the remaining *external features* based on their frequency. These frequently appearing features, together with the features from the first step, to construct the n_2 *external features.*

Finally, the feature space for clustering is constructed with the combination of *original features* and the collected *external features.*

7. EVALUATION

In this Section, we present empirical evaluation results to assess the effectiveness of our proposed framework for short texts clustering. In particular, we conduct experiments on two datasets using six different text representation methods and the results show that our approach is more effective than the state-of-the-art methods. Moreover, some factors that appear to affect the experiment are further discussed.

[5] For more details about "AND" and "OR" query syntax, please refer to http://wiki.apache.org/solr /SolrQuerySyntax

Table 1: Statistics of query length from Google Trends during Nov. 26th 2007 – Nov. 25th 2008

query length	One	Two	Three	more
count	4552	19762	6992	5290
percentage	12.4%	54.0%	19.1%	14.5%

Table 2: The selected hot queries in Web Dataset

NFL	Amazing Grace
Green Bay	Fox News Channel
60 Minutes	New York Giants
Total Eclipse	The Dark Knight
Black Friday	National Economic Council

7.1 Data Sets

Since the average length of 1000 snippets crawled from the Web is 21.72, we define texts that contain no more than 50 words (including stop words) as short texts. To evaluate our methods in web applications, two test collections are employed in our experiment as the benchmark datasets.

Reuters-21578[6]: We remove the texts which contain more than 50 words and filter those clusters with less than 5 texts or more than 500 texts. Thus it leaves 19 clusters comprising 879 texts. The number of texts in each cluster ranges from 6 (the cluster "income") to 438 (the cluster "acq").

Web Dataset: This dataset is built to simulate a real web application. As the users' interests are varied, we choose queries of different length according to the statistics of Google Trends[7] during Nov. 26th 2007 to Nov. 25th 2008, as shown in Table 1. Ten hot queries of diverse topics are selected from Google Trends according to the percentage of query length. The selected queries are as shown in Table 2. Top 100 snippets for each query are retrieved via GoogleAPI to build a 10-category Web Dataset with 1000 texts.

7.2 Clustering Methods and Evaluation Criteria

In this experiment, we employ a clustering package from the freely available machine learning software Weka3 [29] and the number of clusters are predefined. Two clustering algorithms, *K-means* and *EM* are employed in this study to verify the effectiveness of six different text representation methods, as defined below:

- *BOW* (baseline 1) : Traditional "bag of words" model with the *tf-idf* weighting schema.
- *BOW+WN* (baseline 2) : *BOW* integrated with additional features from WordNet as presented in [14].
- *BOW+Wiki* (baseline 3) : *BOW* integrated with additional features from Wikipedia as presented in [1].
- *BOW+Know* (baseline 4) : *BOW* integrated with additional features from Wikipedia and WordNet as in baselines 2 and 3. Feature selection strategy as presented in Section 6 was employed.

Table 3: Average Accuracy test condition

	Same Class	Different Class
Same Cluster	$TruePositive$	$FalsePositive$
Different Cluster	$FalseNegative$	$TrueNegative$

- *BOF* : The bag of *original features* extracted with the hierarchical view we described in Section 4.
- *SemKnow* : Our proposed framework.

We evaluate the performance using F_1*measure* and *Average Accuracy.*

$\mathbf{F_1}$measure: A combination of both *precision* and *recall* that measures the extent to which a cluster contains only objects of a particular class and all objects of that class.

Average Accuracy: A statistical measure of how well a classification test correctly identifies or excludes a condition Table is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{10}$$

where TP, TN, FP, FN are defined in Table 3.

In Table 3, TP (true positive) denotes that two texts are manually labeled with the same class and clustered into same cluster; FN (false negative) denotes that two texts are manually labeled with different classes but clustered into same one. TN (true negative) and FP (false positive) are defined in a similar manner.

7.3 Parameter Setting

As mentioned in Section 5.1, we retrieved top C Wikipedia pages to compute the semantic similarity between two phrases. Given the large scale of Wikipedia corpus and the ranking schema provided by Solr search engine, our experimental result is not sensitive to the number of retrieved documents [6]. Similarly we found the performance is independent of the value of the top ω Wikipedia pages used in Section 5.3. As a result, we empirically set the value $C = 100$ and $\omega = 20$ in our experiment.

In Equation 5, *WikiDice*, *WikiJaccard* and *WikiOverlap* were combined to measure the similarity between two phrases using equal weights. We conducted extensive experiments on different parameter settings and found that assigning equal weights to combine the measures always show the best results. We conjecture that it is because these three measures are all based on the occurrences of phrases in Wikipedia and the correlation coefficients of the measures are similar [2]. Therefore, we set $\alpha = \beta = \frac{1}{3}$ in our experiments.

As discussed in Section 6, θ controls the influence of *external features* to the whole feature space, which is crucial in our experiment. We empirically set θ as 0.5 for *SemKnow*, which means that the *external features* have the same number as the *original features.* The effect of external features in the experiment will be discussed in Section 7.5.

7.4 Performance Evaluation

Experimental results of the six text representation methods on both data collections using the two clustering algorithms are respectively reported in Table 4 and Table 5. In the Tables, AveAccuracy denotes the Average Accuracy for each method and Impr represents the percentage improvement of the methods as compared to *BOW* method. In the

[6] http://daviddlewis.com/resources/testcollections/reuters21578
[7] http://www.google.com/trends

Table 4: F$_1$measure and Average Accuracy results using *k-means* algorithm

	Reuters-21578		Web Dataset	
	F_1measure (Impr)	AveAccuracy (Impr)	F_1measure (Impr)	AveAccuracy (Impr)
BOW	0.471 (N.A.)	0.550 (N.A.)	0.491 (N.A.)	0.563 (N.A.)
BOW + WN	0.473 (+0.43%)	0.552 (+0.26%)	0.530 (+8.01%)	0.576 (+2.30%)
BOW + Wiki	0.481 (+2.03%)	0.563 (+2.18%)	0.556 (+13.38%)	0.584 (+3.85%)
BOW + Know	0.489 (+3.75%)	0.566 (+2.86%)	0.558 (+13.79%)	0.583 (+3.70%)
BOF	0.473 (+0.33%)	0.551 (+0.19%)	0.520 (+5.95%)	0.570 (+1.24%)
SemKnow	**0.497 (+5.41%)**	**0.572 (+3.98%)**	**0.583(+18.81%)**	**0.586(+4.11%)**

Table 5: F$_1$measure and Average Accuracy results using *EM* algorithm

	Reuters-21578		Web Dataset	
	F_1measure (Impr)	AveAccuracy (Impr)	F_1measure (Impr)	AveAccuracy (Impr)
BOW	0.516 (N.A.)	0.579 (N.A.)	0.521 (N.A.)	0.608 (N.A.)
BOW + WN	0.525 (+1.72%)	0.585 (+0.99%)	0.540 (+3.59%)	0.626 (+3.02%)
BOW + Wiki	0.540 (+4.74%)	0.598 (+3.39%)	0.550 (+5.50%)	0.629 (+3.44%)
BOW + Know	0.542 (+5.13%)	0.607 (+4.54%)	0.556 (+6.74%)	0.635 (+4.41%)
BOF	0.520 (+0.82%)	0.594 (+2.63%)	0.536 (+2.73%)	0.624 (+2.55%)
SemKnow	**0.548 (+6.28%)**	**0.622 (+7.51%)**	**0.569 (+9.07%)**	**0.670 (+10.20%)**

experiment, each result denotes an average of 10 test runs by randomly choosing the initial parameters for the clustering method.

From Tables 4 and 5, we observe that *BOW+WN*, *BOW+Wiki* and *BOW+Know* augment the performance of *BOW* model on both datasets using the two clustering algorithms. The performance of *BOW* is improved by incorporating *external features* generated from WordNet, Wikipedia and their combination respectively. It demonstrates the benefit and potential of integrating world knowledge into the representation of short texts. *BOW+Wiki* achieves much better performance than *BOW+WN*; we conjecture that it is because of the huge up-to-date concept space in Wikipedia, which is more appropriate for the snippets that occurred recently. *BOW+Know* further improves the performance over *BOW+Wiki*, demonstrates the benefit of judicious integration of both Wikipedia and WordNet as knowledge sources.

We note that *BOF* also achieves better performance as compared with *BOW* model. We believe that this is because of the integration of internal semantics from hierarchical resolution of the original text. It, however, achieves the least augments with respect to *Baseline* as compared to *BOW+WN*, *BOW+Wiki* and *BOW+Know*. We believe that it is mainly due to the lack of external knowledge.

Comparing our proposed *SemKnow* with the other five methods, it achieves the best F_1measure and Average Accuracy scores on both datasets using *k-means* and *EM* clustering algorithms. The highest improvement with respect to *BOW* is obtained on the Web Dataset using *k-means* algorithm. We apply t-test to compare the performance between *SemKnow* and all the *Baselines*. The results demonstrate that our method significantly outperforms the state-of-the-art methods with the *p-value* < 0.01. Among the methods that integrate semantic concepts from Wikipedia and WordNet, *SemKnow* obtains superior performance as compared to *BOW+Know*. We believe that the improvement stems from the ability of the hierarchical structure in better utilizing the characteristics of features at different levels to generate high quality semantics for clustering.

7.5 Effect of External Features

In order to better understand the effect of *external features*, we conduct experiments using *BOF* with different sizes of *external features* on both datasets. As clustering is unsupervised, it is difficult to obtain prior knowledge. Thus, we evaluate the clustering performance by setting the fraction of *external features* θ from 0 to 1 at increments of 0.1. The results are presented in Figures 4 and 5. The *BOF* line depicts the performance of "bag of *original features*", while the *SemKnow* curve shows the performance of our proposed framework which integrates both *original features* and *external features*.

Figures 4 and 5 depict the results on the Reuters and Web Dataset using *k-means* clustering algorithm. In Figure 4, the curve of *SemKnow* reaches a peak at $\theta = 0.2$. As the fraction of *external features* grows, the performance of *SemKnow* declines. When $\theta > 0.5$, *SemKnow* performs worse than *BOF*, which means the external semantics begin to bring in negative impact on the feature space. In Figure 5, the *SemKnow* achieves the best performance at $\theta = 0.3$ and the trend of curve is similar to that in Figure 4. When the fraction is close to 1, the performance of *SemKnow* is close to *BOF*. Similar phenomena have been observed for *EM* algorithm; we omit the results owing to lack of space.

In general, *SemKnow* achieve the best performance at a small *external feature* set size of $\theta = 0.2$ or $\theta = 0.3$. The possible explanation should be the increasing noise accompanied with the larger *external feature* set. This result also demonstrates the utility of external feature selection strategy. As seen from the figures, *SemKnow* arrives at the best performance with a different fraction of *external features* on different datasets, while the clustering algorithm does not appear to be a major factor to the overall performance, we can infer that different data distributions influence the performance of our framework.

In Table 6, we summarize the performance of *BOW* and our method with optimal parameter θ. As compared to *BOW*, we can achieve 12.35% and 30.39% improvement us-

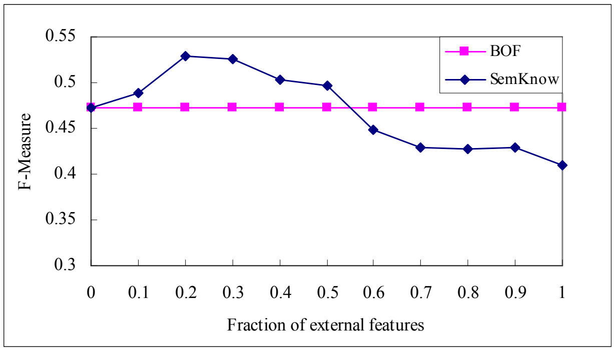

Figure 4: Impact of the parameter θ on Reuters using *K-means* algorithm

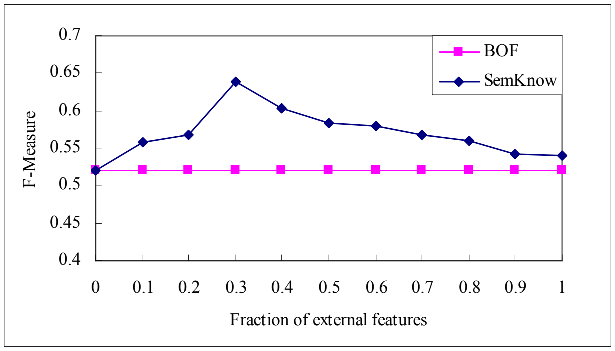

Figure 5: Impact of the parameter θ on Web Dataset using *K-means* algorithm

ing *k-means*, 15.40% and 16.56% improvement using *EM*, on the two datasets respectively. The improvement of optimal θ on *SemKnow* is much higher than that in Tables 4 and 5 with respect to *BOW*. It infers that although the average performance of our method is satisfactory, it can be greatly improved by optimizing the important parameter θ. Therefore, it is possible to manually label some data to optimize the parameter settings when the application has higher requirements in clustering accuracy.

7.6 Detail Analysis

To further analyze the reasons why our proposed method augments the performance of clustering as compared with the previous methods, we illustrate with the *feature space* for the snippet of Figure 1, as shown in Table 7. In the Table, the features on the upper section of the first row are from *word level* of internal semantics, and those in the lower section of first row are from *phrase level*. The external features with dagger are explicit concepts available in Wikipedia articles and features with star are mined from the plain texts of Wikipedia articles. As compared to the previous methods, our framework further removes the feature "likely" from the *word level* features because it is not NP nor VP chunks. This feature is not stop-word but is apparently meaningless for text representation. Several concepts, such as the date ("July 18 2008"), director ("Christopher Nolan"), name of the movie ("The Dark Knight") and comments ("revelatory visceral"), are extracted to construct the original features and they can better tackle the polysemy problem as discussed in Section 4.2. Meanwhile, the external features, especially the features generated by the full extraction of Wikipedia such as the Internet movie database ("IMDB"), title-role in the movie ("joker") and another name of the movie ("batman") handle the synonymy from lexical and semantic aspects well as discussed in Section 5.3. With better handling of the two impediments, namely synonymy and polysemy in NLP, our method achieves satisfactory performance in the clustering of short texts.

Table 6: Optimal results using two algorithms

kmeans	**Reuters**	F_1meas(Impr)	AveAcc(Impr)
	BOW	0.471(N.A.)	0.550(N.A.)
	Optimal	**0.530(+12.35%)**	**0.604(+9.72%)**
	Webdata		
	BOW	0.491(N.A.)	0.563(N.A.)
	Optimal	**0.640(+30.39%)**	**0.607(+7.83%)**
EM	**Reuters**	F_1meas(Impr)	AveAcc(Impr)
	BOW	0.516(N.A.)	0.579(N.A.)
	Optimal	**0.578(+12.02%)**	**0.672(+15.40%)**
	Webdata		
	BOW	0.521(N.A.)	0.608(N.A.)
	Optimal	**0.602(+16.14%)**	**0.709(+16.56%)**

Table 7: Feature space for the snippet in Figure 1

	Feature Space
Internal Semantics	July, 18, 2008, film, year, remain, Christopher, Nolan, Dark, Knight, revelatory, visceral,
	July 18 2008, the best American film, the year, to remain, Christopher Nolan, The Dark Knight, revelatory visceral,
External Semantics	lunar phase†, IMDB*, twelvemonth, to stay, cinema*, batman begins†, batman†, joker*, fireworks*,

8. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel framework to augment the clustering accuracy of short texts by exploiting the internal and external semantics. The internal semantics are extracted by resolving the original texts with a three-level hierarchical view and the external semantics are built with the explicit and implicit concepts derived from multiple semantic knowledge bases. The combination of internal and external semantics well tackled the problems of data sparseness and semantic gap in short texts. Empirical evaluations demonstrated that our framework significant outperformed all the baselines including previously proposed knowledge-based short text clustering methods on two datasets.

There are a number of interesting extensions of this work. As this work is for aggregated search, the efficiency of the whole framework should be optimized for real applications. Moreover, we will explore more tasks in NLP and information retrieval using the internal and external semantics generated by our proposed framework.

9. REFERENCES

[1] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *Proceedings of the 30th ACM SIGIR*, pages 787–788, 2007.

[2] H.-H. Chen, M.-S. Lin, and Y.-C. Wei. Novel association measures using web search with double checking. In *Proceedings of the 21st COLING and the 44th ACL*, pages 1009–1016, 2006.

[3] H. Chim and X. Deng. Efficient phrase-based document similarity for clustering. *IEEE Trans. on Knowl. and Data Eng.*, 20(9):1217–1229, 2008.

[4] M. Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th ACL and Eighth EACL*, pages 16–23, 1997.

[5] D. R. Cutting, D. R. Karger, and J. O. Pedersen. Constant interaction-time scatter/gather browsing of very large document collections. In *Proceedings of the 16th ACM SIGIR*, pages 126–134, 1993.

[6] B. Danushka, M. Yutaka, and I. Mitsuru. Measuring semantic similarity between words using web search engines. In *Proceedings of the 16th WWW*, pages 757–766, 2007.

[7] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th WWW*, pages 519–528, 2003.

[8] L. Denoyer and P. Gallinari. The wikipedia xml corpus. *SIGIR Forum*, 40(1):64–69, 2006.

[9] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the 20th AAAI*, volume 21, pages 1048–1153, 2005.

[10] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st AAAI*, pages 1301–1306, 2006.

[11] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th IJCAI*, pages 6–12, 2007.

[12] J. Hammerton, M. Osborne, S. Armstrong, and W. Daelemans. Introduction to special issue on machine learning approaches to shallow parsing. *Machine Learning Research*, 2:551–558, 2002.

[13] M. Hearst and J. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of the 19th ACM SIGIR*, pages 76–84, 1996.

[14] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

[15] J. Hu, L. Fang, Y. Cao, H. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *Proceedings of the 31st ACM SIGIR*, pages 179–186, 2008.

[16] F. Keller, M. Lapata, and O. Ourioupina. Using the web to overcome data sparseness. In *Proceedings of the 40th ACL*, pages 230–237, 2002.

[17] U. S. Kohomban and W. S. Lee. Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd ACL*, pages 34–41, 2005.

[18] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th ACM SIGIR*, pages 297–304, 2004.

[19] D. Lewis and W. Croft. Term clustering of syntactic phrases. In *Proceedings of the 13th ACM SIGIR*, pages 385–404, 1989.

[20] T. Marinis. Psycholinguistic techniques in second language acquisition research. *Second Language Research*, 19(2):144, 2003.

[21] D. Metzler, S. Dumais, and C. Meek. Similarity measures for short segments of text. *Lecture Notes in Computer Science*, 4425:16, 2007.

[22] S. Osinski, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of the IIS: IIPWM'04 Conference*, page 359, 2004.

[23] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceeding of the 17th WWW*, pages 91–100, 2008.

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

[25] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th WWW*, pages 377–386. ACM New York, NY, USA, 2006.

[26] M. Sushmita, S.;Lalmas. Using digest pages to increase user result space: Preliminary designs. In *SIGIR Workshop on Aggregated Search*, 2008.

[27] E. Terra and C. Clarke. Frequency estimates for statistical word similarity measures. In *Proceedings of HLT/NAACL 2003*, pages 244–251, 2003.

[28] L. Urena-Lopez, M. Buenaga, and J. Gomez. Integrating linguistic resources in TC through WSD. *Computers and the Humanities*, 35(2):215–230, 2001.

[29] I. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques, 2005.

[30] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks-the International Journal of Computer and Telecommunications Networking*, 31(11):1361–1374, 1999.

[31] H. Zeng, Q. He, Z. Chen, W. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of the 27th ACM SIGIR*, pages 210–217, 2004.

[32] T. Zesch, C. Muller, and I. Gurevych. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of LREC*, 2008.

[33] C. Zhang, N. Sun, X. Hu, T. Huang, and T.-S. Chua. Query segmentation based on eigenspace similarity. In *Proceedings of the ACL-IJCNLP 2009 Conference*, pages 185–188, Suntec, Singapore, August 2009.