Clustering to Find Exemplar Terms for Keyphrase Extraction

Zhiyuan Liu, Peng Li, Yabin Zheng, Maosong Sun
Department of Computer Science and Technology
State Key Lab on Intelligent Technology and Systems
National Lab for Information Science and Technology
Tsinghua University, Beijing 100084, China
{lzy.thu, pengli09, yabin.zheng}@gmail.com, sms@tsinghua.edu.cn

Abstract

Keyphrases are widely used as a brief summary of documents. Since manual assignment is time-consuming, various unsupervised ranking methods based on importance scores are proposed for keyphrase extraction. In practice, the keyphrases of a document should not only be statistically important in the document, but also have a good coverage of the document. Based on this observation, we propose an unsupervised method for keyphrase extraction. Firstly, the method finds exemplar terms by leveraging clustering techniques, which guarantees the document to be semantically covered by these exemplar terms. Then the keyphrases are extracted from the document using the exemplar terms. Our method outperforms sate-of-the-art graph-based ranking methods (TextRank) by 9.5% in F1-measure.

1 Introduction

With the development of Internet, information on the web is emerging exponentially. How to effectively seek and manage information becomes an important research issue. Keyphrases, as a brief summary of a document, provide a solution to help organize, manage and retrieve documents, and are widely used in digital libraries and information retrieval.

Keyphrases in articles of journals and books are usually assigned by authors. However, most articles on the web usually do not have human-assigned keyphrases. Therefore, automatic keyphrase extraction is an important research task. Existing methods can be divided into supervised and unsupervised approaches.

The supervised approach (Turney, 1999) regards keyphrase extraction as a classification task. In this approach, a model is trained to determine whether a candidate term of the document is a keyphrase, based on statistical and linguistic features. For the supervised keyphrase extraction approach, a document set with human-assigned keyphrases is required as training set. However, human labelling is time-consuming. Therefore, in this study we focus on unsupervised approach.

As an example of an unsupervised keyphrase extraction approach, the graph-based ranking (Mihalcea and Tarau, 2004) regards keyphrase extraction as a ranking task, where a document is represented by a term graph based on term relatedness, and then a graph-based ranking algorithm is used to assign importance scores to each term. Existing methods usually use term cooccurrences within a specified window size in the given document as an approximation of term relatedness (Mihalcea and Tarau, 2004).

As we know, none of these existing works gives an explicit definition on what are appropriate keyphrases for a document. In fact, the existing methods only judge the importance of each term, and extract the most important ones as keyphrases.

From the observation of human-assigned keyphrases, we conclude that good keyphrases of a document should satisfy the following properties:

1. **Understandable**. The keyphrases are understandable to people. This indicates the extracted keyphrases should be grammatical. For example, “machine learning” is a grammatical phrase, but “machine learned” is not.

2. **Relevant**. The keyphrases are semantically relevant with the document theme. For example, for a document about “machine learning”, we want the keyphrases all about this theme.

3. **Good coverage**. The keyphrases should

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 257–266,
Singapore, 6-7 August 2009.

cover the whole document well. Suppose we have a document describing "Beijing" from various aspects of "location", "atmosphere" and "culture", the extracted keyphrases should cover all the three aspects, instead of just a partial subset of them.

The classification-based approach determines whether a term is a keyphrase in isolation, which could not guarantee Property 3. Neither does the graph-based approach guarantee the top-ranked keyphrases could cover the whole document. This may cause the resulting keyphrases to be inappropriate or badly-grouped.

To extract the appropriate keyphrases for a document, we suggest an unsupervised clustering-based method. Firstly the terms in a document are grouped into clusters based on semantic relatedness. Each cluster is represented by an exemplar term, which is also the centroid of each cluster. Then the keyphrases are extracted from the document using these exemplar terms.

In this method, we group terms based on semantic relatedness, which guarantees a good coverage of the document and meets Property 2 and 3. Moreover, we only extract the keyphrases in accordance with noun group (chunk) patterns, which guarantees the keyphrases satisfy Property 1.

Experiments show that the clustering-based method outperforms the state-of-the-art graph-based approach on precision, recall and F1-measure. Moreover, this method is unsupervised and language-independent, which is applicable in the web era with enormous information.

The rest of the paper is organized as follows. In Section 2, we introduce and discuss the related work in this area. In Section 3, we give an overview of our method for keyphrase extraction. From Section 4 to Section 7, the algorithm is described in detail. Empirical experiment results are demonstrated in Section 8, followed by our conclusions and plans for future work in Section 9.

2 Related Work

A straightforward method for keyphrase extraction is to select keyphrases according to frequency criteria. However, the poor performance of this method drives people to explore other methods. A pioneering achievement is carried out in (Turney, 1999), as mentioned in Section 1, a supervised machine learning method was suggested in this paper which regards keyphrase extraction as a classification task. In this work, parameterized heuristic rules are combined with a genetic algorithm into a system for keyphrase extraction. A different learning algorithm, Naive Bayes method, is applied in (Frank et al., 1999) with improved results on the same data used in (Turney, 1999). Hulth (Hulth, 2003; Hulth, 2004) adds more linguistic knowledge, such as syntactic features, to enrich term representation, which significantly improves the performance. Generally, the supervised methods need manually annotated training set, which may sometimes not be practical, especially in the web scenario.

Starting with TextRank (Mihalcea and Tarau, 2004), graph-based ranking methods are becoming the most widely used unsupervised approach for keyphrase extraction. The work in (Litvak and Last, 2008) applies HITS algorithm on the word graph of a document under the assumption that the top-ranked nodes should be the document keywords. Experiments show that classification-based supervised method provides the highest keyword identification accuracy, while the HITS algorithm gets the highest F-measure. Work in (Huang et al., 2006) also considers each document as a term graph where the structural dynamics of these graphs can be used to identify keyphrases. Wan and Xiao (Wan and Xiao, 2008b) use a small number of nearest neighbor documents to provide more knowledge to improve graph-based keyphrase extraction algorithm for single document. Motivated by similar idea, Wan and Xiao (Wan and Xiao, 2008a) propose to adopt clustering methods to find a small number of similar documents to provide more knowledge for building word graphs for keyword extraction. Moreover, after our submission of this paper, we find that a method using community detection on semantic term graphs is proposed for keyphrase extraction from multi-theme documents (Grineva et al., 2009). In addition, some practical systems, such as KP-Miner (Elbeltagy and Rafea, 2009), also do not need to be trained on a particular human-annotated document set.

In recent years, a number of systems are developed for extracting keyphrases from web documents (Kelleher and Luz, 2005; Chen et al., 2005), email (Dredze et al., 2008) and some other specific sources, which indicates the importance of keyphrase extraction in the web era. However,

none of these previous works has overall consideration on the essential properties of appropriate keyphrases mentioned in Section 1.

We should also note that, although the precision and recall of most current keyphrase extractors are still much lower compared to other NLP-tasks, it does not indicate the performance is poor because even different annotators may assign different keyphrases to the same document. As described in (Wan and Xiao, 2008b), when two annotators were asked to label keyphrases on 308 documents, the Kappa statistic for measuring inter-agreement among them was only 0.70.

3 Algorithm Overview

The method proposed in this paper is mainly inspired by the nature of appropriate keyphrases mentioned in Section 1, namely *understandable*, semantically *relevant* with the document and *high coverage* of the whole document.

Let's analyze the document describing "Beijing" from the aspects of "location", "atmosphere" and "culture". Under the bag-of-words assumption, each term in the document, except for function words, is used to describe an aspect of the theme. Based on these aspects, terms are grouped into different clusters. The terms in the same cluster are more relevant with each other than with the ones in other clusters. Taking the terms "temperature", "cold" and "winter" for example, they may serve the aspect "atmosphere" instead of "location" or some other aspects when talking about "Beijing".

Based on above description, it is thus reasonable to propose a clustering-based method for keyphrase extraction. The overview of the method is:

1. **Candidate term selection.** We first filter out the stop words and select candidate terms for keyphrase extraction.
2. **Calculating term relatedness.** We use some measures to calculate the semantic relatedness of candidate terms.
3. **Term clustering.** Based on term relatedness, we group candidate terms into clusters and find the exemplar terms of each cluster.
4. **From exemplar terms to keyphrases.** Finally, we use these exemplar terms to extract keyphrases from the document.

In the next four sections we describe the algorithm in detail.

4 Candidate Term Selection

Not all words in a document are possible to be selected as keyphrases. In order to filter out the noisy words in advance, we select candidate terms using some heuristic rules. This step proceeds as follows. Firstly the text is tokenized for English or segmented into words for Chinese and other languages without word-separators. Then we remove the stop words and consider the remaining single terms as candidates for calculating semantic relatedness and clustering.

In methods like (Turney, 1999; Elbeltagy and Rafea, 2009), candidate keyphrases were first found using n-gram. Instead, in this method, we just find the single-word terms as the candidate terms at the beginning. After identifying the exemplar terms within the candidate terms, we extract multi-word keyphrases using the exemplars.

5 Calculating Term Relatedness

After selecting candidate terms, it is important to measure term relatedness for clustering. In this paper, we propose two approaches to calculate term relatedness: one is based on term cooccurrence within the document, and the other by leveraging human knowledge bases.

5.1 Cooccurrence-based Term Relatedness

An intuitive method for measuring term relatedness is based on term cooccurrence relations within the given document. The cooccurrence relation expresses the cohesion relationships between terms.

In this paper, cooccurrence-based relatedness is simply set to the count of cooccurrences within a window of maximum w words in the whole document. In the following experiments, the window size w is set from 2 to 10 words.

Each document can be regarded as a word sequence for computing cooccurrence-based relatedness. There are two types of word sequence for counting term cooccurrences. One is the original word sequence without filtering out any words, and the other is after filtering out the stop words or the words with specified part-of-speech (POS) tags. In this paper we select the first type because each word in the sequence takes important role for measuring term cooccurrences, no matter whether

it is a stop word or something else. If we filter out some words, the term relatedness will not be as precise as before.

In experiments, we will investigate how the window size influences the performance of keyphrase extraction.

5.2 Wikipedia-based Term Relatedness

Many methods have been proposed for measuring the relatedness between terms using external resources. One principled method is leveraging human knowledge bases. Inspired by (Gabrilovich and Markovitch, 2007), we adopt Wikipedia, the largest encyclopedia collected and organized by human on the web, as the knowledge base to measure term relatedness.

The basic idea of computing term relatedness by leveragting Wikipedia is to consider each Wikipedia article as a concept. Then the semantic meaning of a term could be represented as a weighted vector of Wikipedia concepts, of which the values are the term's TFIDF within corresponding Wikipedia articles. We could compute the term relatedness by comparing the concept vectors of the terms. Empirical evaluations confirm that the idea is effective and practical for computing term relatedness (Gabrilovich and Markovitch, 2007).

In this paper, we select cosine similarity, Euclidean distance, Point-wise Mutual Information and Normalized Google Similarity Distance (Cilibrasi and Vitanyi, 2007) for measuring term relatedness based on the vector of Wikipedia concepts.

Denote the Wikipedia-concept vector of the term t_i as $C_i = \{c_{i1}, c_{i2}, ..., c_{iN}\}$, where N indicates the number of Wikipedia articles, and c_{ik} is the TFIDF value of w_i in the kth Wikipedia article. The cosine similarity is defined as

$$cos(i,j) = \frac{C_i \cdot C_j}{\|C_i\|\|C_j\|} \tag{1}$$

The definition of Euclidean distance is

$$euc(i,j) = \sqrt{\sum_{k=1}^{N}(c_{ik} - c_{jk})^2} \tag{2}$$

Point-wise Mutual Information (PMI) is a common approach to quantify relatedness. Here we take three ways to measure term relatedness using PMI. One is based on Wikipedia page count,

$$pmi_p(i,j) = \log_2 \frac{N \times p(i,j)}{p(i) \times p(j)} \tag{3}$$

where $p(i,j)$ is the number of Wikipedia articles containing both t_i and t_j, while $p(i)$ is the number of articles which contain t_i. The second is based on the term count in Wikipedia articles,

$$pmi_t(i,j) = \log_2 \frac{T \times t(i,j)}{t(i) \times t(j)} \tag{4}$$

where T is the number of terms in Wikipedia, $t(i,j)$ is the number of t_i and t_j occurred adjacently in Wikipedia, and $t(i)$ is the number of t_i in Wikipedia. The third one is a combination of the above two PMI ways,

$$pmi_c(i,j) = \log_2 \frac{N \times pt(i,j)}{p(i) \times p(j)} \tag{5}$$

where $pt(i,j)$ indicates the number of Wikipedia articles containing t_i and t_j as adjacency. It is obvious that $pmi_c(i,j) \leq pmi_p(i,j)$, and $pmi_c(i,j)$ is more strict and accurate for measuring relatedness.

Normalized Google Similarity Distance (NGD) is a new measure for measuring similarity between terms proposed by (Cilibrasi and Vitanyi, 2007) based on information distance and Kolmogorov complexity. It could be applied to compute term similarity from the World Wide Web or any large enough corpus using the page counts of terms. NGD used in this paper is based on Wikipedia article count, defined as

$$ngd(i,j) = \frac{\max(\log p(i), \log p(j)) - logp(i,j)}{\log N - \min(logp(i), logp(j))} \tag{6}$$

where N is the number of Wikipedia articles used as normalized factor.

Once we get the term relatedness, we could then group the terms using clustering techniques and find exemplar terms for each cluster.

6 Term Clustering

Clustering is an important unsupervised learning problem, which is the assignment of objects into groups so that objects from the same cluster are more similar to each other than objects from different clusters (Han and Kamber, 2005). In this paper, we use three widely used clustering algorithms, hierarchical clustering, spectral clustering and Affinity Propagation, to cluster the candidate terms of a given document based on the semantic relatedness between them.

6.1 Hierarchical Clustering

Hierarchical clustering groups data over a variety of scales by creating a cluster tree. The tree is a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. The hierarchical clustering follows this procedure:

1. Find the distance or similarity between every pair of data points in the dataset;
2. Group the data points into a binary and hierarchical cluster tree;
3. Determine where to cut the hierarchical tree into clusters. In hierarchical clustering, we have to specify the cluster number m in advance.

In this paper, we use the hierarchical clustering implemented in Matlab Statistics Toolbox. Note that although we use hierarchical clustering here, the cluster hierarchy is not necessary for the clustering-based method.

6.2 Spectral Clustering

In recent years, spectral clustering has become one of the most popular modern clustering algorithms. Spectral clustering makes use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering into fewer dimensions, which is simple to implement and often outperforms traditional clustering methods such as k-means. Detailed introduction to spectral clustering could be found in (von Luxburg, 2006).

In this paper, we use the spectral clustering toolbox developed by Wen-Yen Chen, et al. (Chen et al., 2008) [1]. Since the cooccurrence-based term relatedness is usually sparse, the traditional eigenvalue decomposition in spectral clustering will sometimes get run-time error. In this paper, we use the singular value decomposition (SVD) technique for spectral clustering instead.

For spectral clustering, two parameters are required to be set by the user: the cluster number m, and σ which is used in computing similarities from object distances

$$s(i,j) = \exp(\frac{-d(i,j)^2}{2\sigma^2}) \quad (7)$$

where $s(i,j)$ and $d(i,j)$ are the similarity and distance between i and j respectively.

[1]The package could be accessed via `http://www.cs.ucsb.edu/~wychen/sc.html`.

6.3 Affinity Propagation

Another powerful clustering method, Affinity Propagation, is based on message passing techniques. AP was proposed in (Frey and Dueck, 2007), where AP was reported to find clusters with much lower error than those found by other methods. In this paper, we use the toolbox developed by Frey, et al. [2].

Detailed description of the algorithm could be found in (Frey and Dueck, 2007). Here we introduced three parameters for AP:

- **Preference**. Rather than requiring predefined number of clusters, Affinity Propagation takes as input a real number p for each term, so that the terms with larger p are more likely to be chosen as exemplars, i.e., centroids of clusters. These values are referred to as "preferences". The preferences are usually be set as the maximum, minimum, mean or median of $s(i,j), i \neq j$.
- **Convergence criterion**. AP terminates if (1) the local decisions stay constant for I_1 iterations; or (2) the number of iterations reaches I_2. In this work, we set I_1 to 100 and I_2 to $1,000$.
- **Damping factor**. When updating the messages, it is important to avoid numerical oscillations by using damping factor. Each message is set to λ times its value from the previous iteration plus $1-\lambda$ times its prescribed updated value, where the damping factor λ is between 0 and 1. In this paper we set $\lambda = 0.9$.

7 From Exemplar Terms to Keyphrases

After term clustering, we select the exemplar terms of each clusters as seed terms. In Affinity Propagation, the exemplar terms are directly obtained from the clustering results. In hierarchical clustering, exemplar terms could also be obtained by the Matlab toolbox. While in spectral clustering, we select the terms that are most close to the centroid of a cluster as exemplar terms.

As reported in (Hulth, 2003), most manually assigned keyphrases turn out to be noun groups. Therefore, we annotate the document with POS

[2]The package could be accessed via `http://www.psi.toronto.edu/affinitypropagation/`.

tags using Stanford Log-Linear Tagger [3], and then extract the noun groups whose pattern is zero or more adjectives followed by one or more nouns. The pattern can be represented using regular expressions as follows

$$(JJ) * (NN|NNS|NNP)+$$

where JJ indicates adjectives and various forms of nouns are represented using NN, NNS and NNP. From these noun groups, we select the ones that contain one or more exemplar terms to be the keyphrases of the document.

In this process, we may find single-word keyphrases. In practice, only a small fraction of keyphrases are single-word. Thus, as a part of postprocessing process, we have to use a frequent word list to filter out the terms that are too common to be keyphrases.

8 Experiment Results

8.1 Datasets and Evaluation Metric

The dataset used in the experiments is a collection of scientific publication abstracts from the *Inspec* database and the corresponding manually assigned keyphrases [4]. The dataset is used in both (Hulth, 2003) and (Mihalcea and Tarau, 2004). Each abstract has two kinds of keyphrases: controlled keyphrases, restricted to a given dictionary, and uncontrolled keyphrases, freely assigned by the experts. We use the uncontrolled keyphrases for evaluation as proposed in (Hulth, 2003) and followed by (Mihalcea and Tarau, 2004).

As indicated in (Hulth, 2003; Mihalcea and Tarau, 2004), in uncontrolled manually assigned keyphrases, only the ones that occur in the corresponding abstracts are considered in evaluation. The extracted keyphrases of various methods and manually assigned keyphrases are compared after stemming.

In the experiments of (Hulth, 2003), for her supervised method, Hulth splits a total of $2,000$ abstracts into $1,000$ for training, 500 for validation and 500 for test. In (Mihalcea and Tarau, 2004), due to the unsupervised method, only the test set was used for comparing the performance of TextRank and Hulth's method.

For computing Wikipedia-based relatedness, we use a snapshot on November 11, 2005 [5]. The frequent word list used in the postprocessing step for filtering single-word phrases is also computed from Wikipedia. In the experiments of this paper, we add the words that occur more than $1,000$ times in Wikipedia into the list.

The clustering-based method is completely unsupervised. Here, we mainly run our method on test set and investigate the influence of relatedness measurements and clustering methods with different parameters. Then we compare our method with two baseline methods: Hulth's method and TextRank. Finally, we analyze and discuss the performance of the method by taking the abstract of this paper as a demonstration.

8.2 Influence of Relatedness Measurements

We first investigate the influence of semantic relatedness measurements. By systematic experiments, we find that Wikipedia-based relatedness outperforms cooccurrence-based relatedness for keyphrase extraction, though the improvement is not significant. In Table 1, we list the performance of spectral clustering with various relatedness measurements for demonstration. In this table, the w indicates the window size for counting cooccurrences in cooccurrence-based relatedness. cos, euc, etc. are different measures for computing Wikipedia-based relatedness which we presented in Section 5.2.

Table 1: Influence of relatedness measurements for keyphrase extraction.

Parameters	Precision	Recall	F1-measure
Cooccurrence-based Relatedness			
$w = 2$	0.331	0.626	0.433
$w = 4$	0.333	0.621	0.434
$w = 6$	0.331	0.630	0.434
$w = 8$	0.330	0.623	0.432
$w = 10$	0.333	0.632	0.436
Wikipedia-based Relatedness			
cos	0.348	0.655	0.455
euc	0.344	0.634	0.446
pmi_p	0.344	0.621	0.443
pmi_t	0.344	0.619	0.442
pmi_c	0.350	0.660	**0.457**
ngd	0.343	0.620	0.442

[3] The package could be accessed via `http://http://nlp.stanford.edu/software/tagger.shtml`.

[4] Many thanks to Anette Hulth for providing us the dataset.

[5] The dataset could be get from `http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/`.

We use spectral clustering here because it outperforms other clustering techniques, which will be shown in the next subsection. The results in Table 1 are obtained when the cluster number $m = \frac{2}{3}n$, where n is the number of candidate terms obtained in Section 5. Besides, for Euclidean distance and Google distance, we set $\sigma = 36$ of Formula 7 to convert them to corresponding similarities, where we get the best result when we conduct different trails with $\sigma = 9, 18, 36, 54$, though there are only a small margin among them.

As shown in Table 1, although the method using Wikipedia-based relatedness outperforms that using cooccurrence-based relatedness, the improvement is not prominent. Wikipedia-based relatedness is computed according to global statistical information on Wikipedia. Therefore it is more precise than cooccurrence-based relatedness, which is reflected in the performance of the keyphrase extraction. However, on the other hand, Wikipedia-based relatedness does not catch the document-specific relatedness, which is represented by the cooccurrence-based relatedness. It will be an interesting future work to combine these two types of relatedness measurements.

From this subsection, we conclude that, although the method using Wikipedia-based relatedness performs better than cooccurrence-based one, due to the expensive computation of Wikipedia-based relatedness, the cooccurrence-based one is good enough for practical applications.

8.3 Influence of Clustering Methods and Their Parameters

To demonstrate the influence of clustering methods for keyphrase extraction, we fix the relatedness measurement as Wikipedia-based pmi_c, which has been shown in Section 8.2 to be the best relatedness measurement.

In Table 2, we show the performance of three clustering techniques for keyphrase extraction. For hierarchical clustering and spectral clustering, the cluster number m are set explicitly as the proportion of candidate terms n, while for Affinity Propagation, we set preferences as the minimum, mean, median and maximum of $s(i, j)$ to get different number of clusters, denoted as min, $mean$, $median$ and max in the table respectively.

As shown in the table, when cluster number m is large, spectral clustering outperforms hierarchical clustering and Affinity Propagation. Among these methods, only Affinity Propagation under some parameters performs poorly.

Table 2: Influence of clustering methods for keyphrase extraction.

Parameters	Precision	Recall	F1-measure
Hierarchical Clustering			
$m = \frac{1}{4}n$	0.365	0.369	0.367
$m = \frac{1}{3}n$	0.365	0.369	0.367
$m = \frac{1}{2}n$	0.351	0.562	0.432
$m = \frac{2}{3}n$	0.346	0.629	0.446
$m = \frac{4}{5}n$	0.340	0.657	0.448
Spectral Clustering			
$m = \frac{1}{4}n$	0.385	0.409	0.397
$m = \frac{1}{3}n$	0.374	0.497	0.427
$m = \frac{1}{2}n$	0.374	0.497	0.427
$m = \frac{2}{3}n$	0.350	0.660	**0.457**
$m = \frac{4}{5}n$	0.340	0.679	0.453
Affinity Propagation			
$p = max$	0.331	0.688	0.447
$p = mean$	0.433	0.070	0.121
$p = median$	0.422	0.078	0.132
$p = min$	0.419	0.059	0.103

8.4 Comparing with Other Algorithms

Table 3 lists the results of the clustering-based method compared with the best results reported in (Hulth, 2003; Mihalcea and Tarau, 2004) on the same dataset. For each method, the table lists the total number of assigned keyphrases, the mean number of keyphrases per abstract, the total number of correct keyphrases, and the mean number of correct keyphrases. The table also lists precision, recall and F1-measure. In this table, hierarchical clustering, spectral clustering and Affinity Propagation are abbreviated by "HC", "SC" and "AP" respectively.

The result of Hulth's method listed in this table is the best one reported in (Hulth, 2003) on the same dataset. This is a supervised classification-based method, which takes more linguistic features in consideration for keyphrase extraction. The best result is obtained using n-gram as candidate keyphrases and adding POS tags as candidate features for classification.

The result of TextRank listed here is the best one reported in (Mihalcea and Tarau, 2004) on the same dataset. To obtain the best result, the authors built an undirected graph using window $w = 2$ on word sequence of the given document, and ran

Table 3: Comparison results of Hulth's method, TextRank and our clustering-based method.

	Assigned		Correct				
Method	Total	Mean	Total	Mean	Precision	Recall	F1-measure
Hulth's	7,815	15.6	1,973	3.9	0.252	0.517	0.339
TextRank	6,784	13.7	2,116	4.2	0.312	0.431	0.362
HC	7,303	14.6	2,494	5.0	0.342	0.657	0.449
SC	7,158	14.3	2,505	5.0	**0.350**	0.660	**0.457**
AP	8,013	16.0	2,648	5.3	0.330	**0.697**	0.448

PageRank on it.

In this table, the best result of hierarchical clustering is obtained by setting the cluster number $m = \frac{2}{3}n$ and using Euclidean distance for computing Wikipedia-based relatedness. The parameters of spectral clustering are the same as in last subsection. For Affinity Propagation, the best result is obtained under $p = max$ and using Wikipedia-based Euclidean distance as relatedness measure.

From this table, we can see clustering-based method outperforms TextRank and Hulth's method. For spectral clustering, F1-measure achieves an approximately 9.5% improvement as compared to TextRank.

Furthermore, since the clustering-based method is unsupervised, we do not need any set for training and validation. In this paper, we also carry out an experiment on the whole Hulth's dataset with $2,000$ abstracts. The performance is similar to that on 500 abstracts as shown above. The best result is obtained when we use spectral clustering by setting $m = \frac{2}{3}n$ with Wikipedia-based pmi_c relatedness, which is the same in 500 abstracts. In this result, we extract $29,517$ keyphrases, among which $9,655$ are correctly extracted. The precision, recall and F1-measure are 0.327, 0.653 and 0.436 respectively. The experiment results show that the clustering-based method is stable.

8.5 Analysis and Discussions

From the above experiment results, we can see the clustering-based method is both robust and effective for keyphrase extraction as an unsupervised method.

Here, as an demonstration, we use spectral clustering and Wikipedia-based pmi_c relatedness to extract keyphrases from the abstract of this paper. The extracted stemmed keyphrases under various cluster numbers are shown in Figure 1. In this figure, we find that when $m = \frac{1}{4}n, \frac{1}{3}n, \frac{1}{2}n$, the extracted keyphrases are identical, where the exemplar terms under $m = \frac{1}{3}n$ are marked in boldface. We find several aspects like "unsupervised", "exemplar term" and "keyphrase extraction" are extracted correctly. In fact, "clustering technique" in the abstract should also be extracted as a keyphrase. However, since "clustering" is tagged as a verb that ends in -ing, which disagrees the noun group patterns, thus the phrase is not among the extracted keyphrases.

When $m = \frac{2}{3}n$, the extracted keyphrases are noisy with many single-word phrases. As the cluster number increases, more exemplar terms are identified from these clusters, and more keyphrases will be extracted from the document based on exemplar terms. If we set the cluster number to $m = n$, all terms will be selected as exemplar terms. In this extreme case, all noun groups will be extracted as keyphrases, which is obviously not proper for keyphrase extraction. Thus, it is important for this method to appropriately specify the cluster number.

In the experiments, we also notice that frequent word list is important for keyphrase extraction. Without the list for filtering, the best F1-measure will decrease by about 5 percent to 40%. However, the solution of using frequent word list is somewhat too simple, and in future work, we plan to investigate a better combination of clustering-based method with traditional methods using term frequency as the criteria.

9 Conclusion and Future Work

In this paper, we propose an unsupervised clustering-based keyphrase extraction algorithm. This method groups candidate terms into clusters and identify the exemplar terms. Then keyphrases are extracted from the document based on the exemplar terms. The clustering based on term semantic relatedness guarantees the extracted keyphrases have a good coverage of the document. Experiment results show the method has a good ef-

Figure 1: Keyphrases in stemmed form extracted from this paper's abstract.

Keyphrases when $m = \frac{1}{4}n, \frac{1}{3}n, \frac{1}{2}n$
unsupervis method; various **unsupervis** rank method; **exemplar term**; state-of-the-art **graph-bas** rank method; **keyphras**; **keyphras** extract
Keyphrases when $m = \frac{2}{3}n$
unsupervis method; manual assign; brief summari; various unsupervis rank method; exemplar term; document; state-of-the-art graph-bas rank method; experi; keyphras; import score; keyphras extract

fectiveness and robustness, and outperforms baselines significantly.

Future work may include:

1. Investigate the feasibility of clustering directly on noun groups;

2. Investigate the feasibility of combining cooccurrence-based and Wikipedia-based relatedness for clustering;

3. Investigate the performance of the method on other types of documents, such as long articles, product reviews and news;

4. The solution of using frequent word list for filtering out too common single-word keyphrases is undoubtedly simple, and we plan to make a better combination of the clustering-based method with traditional frequency-based methods for keyphrase extraction.

Acknowledgments

This work is supported by the National 863 Project under Grant No. 2007AA01Z148 and the National Science Foundation of China under Grant No. 60621062. The authors would like to thank Anette Hulth for kindly sharing her datasets.

References

Mo Chen, Jian-Tao Sun, Hua-Jun Zeng, and Kwok-Yan Lam. 2005. A practical system of keyphrase extraction for web pages. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 277–278.

Wen Y. Chen, Yangqiu Song, Hongjie Bai, Chih J. Lin, and Edward Chang. 2008. Psc: Paralel spectral clustering. Submitted.

Rudi L. Cilibrasi and Paul M. B. Vitanyi. 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 199–206.

S. Elbeltagy and A. Rafea. 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1):132–144.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 668–673.

Brendan J J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*.

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12.

M. Grineva, M. Grinev, and D. Lizorkin. 2009. Extracting key terms from noisy and multi-theme documents. In *Proceedings of the 18th international conference on World wide web*, pages 661–670. ACM New York, NY, USA.

Jiawei Han and Micheline Kamber. 2005. *Data Mining: Concepts and Techniques, second edition*. Morgan Kaufmann.

Chong Huang, Yonghong Tian, Zhi Zhou, Charles X. Ling, and Tiejun Huang. 2006. Keyphrase extraction using semantic networks structure analysis. In *Proceedings of the 6th International Conference on Data Mining*, pages 275–284.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223.

A. Hulth. 2004. Reducing false positives by expert combination in automatic keyword indexing. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, page 367.

Daniel Kelleher and Saturnino Luz. 2005. Automatic hypertext keyphrase detection. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*.

Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, pages 17–24.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Peter D. Turney. 1999. Learning to Extract Keyphrases from Text. *National Research Council Canada, Institute for Information Technology, Technical Report ERB-1057*.

U. von Luxburg. 2006. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics.

Xiaojun Wan and Jianguo Xiao. 2008a. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of COLING*, pages 969–976.

Xiaojun Wan and Jianguo Xiao. 2008b. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 855–860.