CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction

Xiaojun Wan and Jianguo Xiao
Institute of Computer Science and Technology
Peking University, Beijing 100871, China
{wanxiaojun,xiaojianguo}@icst.pku.edu.cn

Abstract

Previous methods usually conduct the keyphrase extraction task for single documents separately without interactions for each document, under the assumption that the documents are considered independent of each other. This paper proposes a novel approach named CollabRank to collaborative single-document keyphrase extraction by making use of mutual influences of multiple documents within a cluster context. CollabRank is implemented by first employing the clustering algorithm to obtain appropriate document clusters, and then using the graph-based ranking algorithm for collaborative single-document keyphrase extraction within each cluster. Experimental results demonstrate the encouraging performance of the proposed approach. Different clustering algorithms have been investigated and we find that the system performance relies positively on the quality of document clusters.

1 Introduction

A keyphrase is defined as a meaningful and significant expression consisting of one or more words in a document. Appropriate keyphrases can be considered as a highly condensed summary for a document, and they can be used as a label for the document to supplement or replace the title or summary, thus facilitating users' fast browsing and reading. Moreover, document keyphrases have been successfully used in the following IR and NLP tasks: document indexing (Gutwin et al., 1999), document classification (Krulwich and Burkey, 1996), document clustering (Zhang et al., 2004; Hammouda et al., 2005) and document summarization (Berger and Mittal, 2000; Buyukkokten et al., 2001).

Keyphrases are usually manually assigned by authors, especially for journal or conference articles. However, the vast majority of documents (e.g. news articles, magazine articles) do not have keyphrases, therefore it is beneficial to automatically extract a few keyphrases from a given document to deliver the main content of the document. Here, keyphrases are selected from within the body of the input document, without a predefined list (i.e. controlled vocabulary). Most previous work focuses on keyphrase extraction for journal or conference articles, while this paper focus on keyphrase extraction for news articles because news article is one of the most popular document genres on the web and most news articles have no author-assigned keyphrases.

Very often, keyphrases of all single documents in a document set are required to be extracted. However, all previous methods extract keyphrases for a specified document based only on the information contained in that document, such as the phrase's TFIDF, position and other syntactic information in the document. One common assumption of existing methods is that the documents are independent of each other. Hence the keyphrase extraction task is conducted separately without interactions for each document. However, the multiple documents within an appropriate cluster context usually have mutual influences and contain useful clues which can help to extract keyphrases from each other. For example, two documents about the same topic "earthquake" would share a few common phrases, e.g. "earthquake", "victim", and they can provide additional knowledge for each other to better evaluate and extract salient keyphrases from each other. The idea is borrowed from human's perception that a user would better understand a topic expressed in a document if the user reads more documents about the same topic.

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 969–976
Manchester, August 2008

Based on the above assumption, we propose a novel framework for collaborative single-document keyphrase extraction by making use of the additional information from multiple documents within an appropriate cluster context. The collaborative framework for keyphrase extraction consists of the step of obtaining the cluster context and the step of collaborative keyphrase extraction in each cluster. In this study, the cluster context is obtained by applying the clustering algorithm on the document set, and we have investigated how the cluster context influences the keyphrase extraction performance by employing different clustering algorithms. The graph-based ranking algorithm is employed for collaborative keyphrase extraction for each document in a specified cluster. Instead of making only use of the word relationships in a single document, the algorithm can incorporate the "voting" or "recommendations" between words in all the documents of the cluster, thus making use of the global information existing in the cluster context. The above implementation of the collaborative framework is denoted as CollabRank in this paper.

Experiments have been performed on a dataset consisting of 308 news articles with human-annotated keyphrases, and the results demonstrate the good effectiveness of the CollabRank approach. We also find that the extraction performance is positively correlated with the quality of cluster context, and existing clustering algorithms can yield appropriate cluster context for collaborative keyphrase extraction.

The rest of this paper is organized as follows: Section 2 introduces the related work. The proposed CollabRank is described in detail in Section 3. Empirical evaluation is demonstrated in Section 4 and lastly we conclude this paper in Section 5.

2 Related Work

The methods for keyphrase (or keyword) extraction can be roughly categorized into either unsupervised or supervised.

Unsupervised methods usually involve assigning a saliency score to each candidate phrases by considering various features. Krulwich and Burkey (1996) use heuristics based on syntactic clues to extract keyphrases from a document. Barker and Cornacchia (2000) propose a simple system for choosing noun phrases from a document as keyphrases. Muñoz (1996) uses an unsupervised learning algorithm to discover two-word keyphrases. The algorithm is based on Adaptive Resonance Theory (ART) neural networks. Steier and Belew (1993) use the mutual information statistics to discover two-word keyphrases. Tomokiyo and Hurst (2003) use pointwise KL-divergence between multiple language models for scoring both phraseness and informativeness of phrases. More recently, Mihalcea and Tarau (2004) propose the TextRank model to rank keywords based on the co-occurrence links between words. Such algorithms make use of "voting" or "recommendations" between words to extract keyphrases.

Supervised machine learning algorithms have been proposed to classify a candidate phrase into either keyphrase or not. GenEx (Turney, 2000) and Kea (Frank et al., 1999; Witten et al., 1999) are two typical systems, and the most important features for classifying a candidate phrase are the frequency and location of the phrase in the document. More linguistic knowledge has been explored by Hulth (2003). Statistical associations between keyphrases have been used to enhance the coherence of the extracted keyphrases (Turney, 2003). Song et al. (2003) present an information gain-based keyphrase extraction system called KPSpotter. Medelyan and Witten (2006) propose KEA++ that enhances automatic keyphrase extraction by using semantic information on terms and phrases gleaned from a domain-specific thesaurus. Nguyen and Kan (2007) focus on keyphrase extraction in scientific publications by using new features that capture salient morphological phenomena found in scientific keyphrases.

The tasks of keyphrase extraction and document summarization are similar and thus they have been conducted in a uniform framework. Zha (2002) proposes a method for simultaneous keyphrase extraction and text summarization by using the heterogeneous sentence-to-word relationships. Wan et al. (2007a) propose an iterative reinforcement approach to simultaneous keyphrase extraction and text summarization. Other related works include web page keyword extraction (Kelleher and Luz, 2005; Zhang et al., 2005; Chen et al., 2005), advertising keywords finding (Yih et al., 2006).

To the best of our knowledge, all previous work conducts the task of keyphrase extraction for each single document independently, without making use of the collaborative knowledge in multiple documents. We focus on unsupervised methods in this study.

3 The Proposed CollabRank Approach

3.1 Framework Description

Given a document set for keyphrase extraction of each single document, CollabRank first employs the clustering algorithm to group the documents into a few clusters. The documents within each cluster are expected to be topic-related and each cluster can be considered as a context for any document in the cluster. Given a document cluster, CollabRank makes use of the global word relationships in the cluster to evaluate and rank candidate phrases for each single document in the cluster based on the graph-based ranking algorithm. Figure 1 gives the framework of the proposed approach.

1. **Document Clustering**: *Group the documents in the document set D into a few clusters using the clustering algorithm;*
2. **Collaborative Keyphrase Extraction**: *For each cluster C, perform the following steps respectively to extract keyphrases for single documents in the cluster in a batch mode:*

   1) **Cluster-level Word Evaluation***: Build a global affinity graph G based on all candidate words restricted by syntactic filters in the documents of the given cluster C, and employ the graph-ranking based algorithm to compute the cluster-level saliency score for each word.*

   2) **Document-level Keyphase Extraction***: For any single document d in the cluster, evaluate the candidate phrases in the document based on the scores of the words contained in the phrases, and finally choose a few phrases with highest scores as the keyphrases of the document.*

Figure 1. The Framework of CollabRank

In the first step of the above framework, different clustering algorithms will yield different clusters. The documents in a high-quality cluster are usually deemed to be highly topic-related (i.e. appropriate cluster context), while the documents in a low-quality cluster are usually not topic-related (i.e. inappropriate cluster context). The quality of a cluster will influence the reliability of the contextual information for evaluating the words in the cluster. A number of clustering algorithms will be investigated in the experiments, including the agglomerative algorithm (both average-link and complete-link), the divisive algorithm, and the kmeans algorithm (Jain et al., 1999), whose details will be described in the evalution section.

In the second step of the above framework, substep 1) aims to evaluate all candidate words in the cluster based on the graph-based ranking algorithm. The global affinity graph aims to reflect the cluster-level co-occurrence relationships between all candidate words in the documents of the given cluster. The saliency scores of the words are computed based on the global affinity graph to indicate how much information about the main topic the words reflect. Substep 2) aims to evaluate candidate phrases of each single document based on the cluster-level word scores, and then choose a few salient phrases as keyphrases of the document. Substep 1) is performed on all documents in the cluster in order to evaluate the words from a global perspective, while substep 2) is performed on each single document in order to extract keyphrases from a local perspective. A keyphrase of a document is expected to include highly salient words. We can see that the keyphrase extraction tasks are conducted in a batch mode for each cluster. The substeps of 1) and 2) will be described in next sections respectively. If substep 1) is performed on each single document without considering the cluster context, the approach is degenerated into the simple TextRank model (Mihalcea and Tarau, 2004), which is denoted as SingleRank in this paper.

It is noteworthy that in addition to the graph-based ranking algorithm, other keyphrase extraction methods can also be integrated in the proposed collaborative framework to exploit the collaborative knowledge in the cluster context.

3.2 Cluster-Level Word Evaluation

Like the PageRank algorithm (Page et al., 1998), the graph-based ranking algorithm employed in this study is essentially a way of deciding the importance of a vertex within a graph based on global information recursively drawn from the entire graph. The basic idea is that of "voting" or "recommendation" between the vertices. A link between two vertices is considered as a vote cast from one vertex to the other vertex. The score associated with a vertex is determined by the votes that are cast for it, and the score of the vertices casting these votes.

Formally, given a specified cluster C, let $G=(V, E)$ be an undirected graph to reflect the relationships between words in the cluster. V is the set of vertices and each vertex is a candidate word[2] in the cluster. Because not all words in the documents are good indicators of keyphrases, the words added to the graph are restricted with syntactic filters, i.e., only the words with a certain part of speech are added. As in Mihalcea and Tarau (2004), the documents are tagged by a

[2] The original words are used without stemming.

POS tagger, and only the nouns and adjectives are added into the vertex set[3]. E is the set of edges, which is a subset of $V \times V$. Each edge e_{ij} in E is associated with an affinity weight $aff(v_i,v_j)$ between words v_i and v_j. The weight is computed based on the co-occurrence relation between the two words, controlled by the distance between word occurrences. The co-occurrence relation can express cohesion relationships between words. Two vertices are connected if the corresponding words co-occur at least once within a window of maximum k words, where k can be set anywhere from 2 to 20 words. The affinity weight $aff(v_i,v_j)$ is simply set to be the count of the controlled co-occurrences between the words v_i and v_j in the whole cluster as follows:

$$aff(v_i,v_j) = \sum_{d \in C} count_d(v_i,v_j) \quad (1)$$

where $count_d(v_i,v_j)$ is the count of the controlled co-occurrences between words v_i and v_j in document d.

The graph is built based on the whole cluster and it is called *Global Affinity Graph*. The biggest difference between CollabRank and SingleRank is that SingleRank builds a local graph based on each single document.

We use an affinity matrix M to describe G with each entry corresponding to the weight of an edge in the graph. $M = (M_{i,j})_{|V| \times |V|}$ is defined as follows:

$$M_{i,j} = \begin{cases} aff(v_i,v_j), & \text{if } v_i \text{ links with } v_j \text{ and } i \neq j; \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Then M is normalized to $\tilde{M}$ as follows to make the sum of each row equal to 1:

$$\tilde{M}_{i,j} = \begin{cases} M_{i,j} \Big/ \sum_{j=1}^{|V|} M_{i,j}, & \text{if } \sum_{j=1}^{|V|} M_{i,j} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Based on the global affinity graph G, the cluster-level saliency score $WordScore_{clus}(v_i)$ for word v_i can be deduced from those of all other words linked with it and it can be formulated in a recursive form as in the PageRank algorithm:

$$WordScore_{clus}(v_i) = \mu \cdot \sum_{all\ j \neq i} WordScore_{clus}(v_j) \cdot \tilde{M}_{j,i} + \frac{(1-\mu)}{|V|} \quad (4)$$

And the matrix form is:

$$\vec{\lambda} = \mu \tilde{M}^T \vec{\lambda} + \frac{(1-\mu)}{|V|} \vec{e} \quad (5)$$

where $\vec{\lambda} = [WordScore_{clus}(v_i)]_{|V| \times 1}$ is the vector of word saliency scores. $\vec{e}$ is a vector with all elements equaling to 1. μ is the damping factor usually set to 0.85, as in the PageRank algorithm.

The above process can be considered as a Markov chain by taking the words as the states and the corresponding transition matrix is given by $\mu \tilde{M}^T + \frac{(1-\mu)}{|V|} \vec{e}\vec{e}^T$. The stationary probability distribution of each state is obtained by the principal eigenvector of the transition matrix.

For implementation, the initial scores of the words are set to 1 and the iteration algorithm in Equation (4) is adopted to compute the new scores of the words. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any words falls below a given threshold (0.0001 in this study).

For SingleRank, the saliency score $WordScore_{doc}(v_i)$ for word v_i is computed in the same iterative way based on the local graph for the single document.

3.3 Document-Level Keyphrase Extraction

After the scores of all candidate words in the cluster have been computed, candidate phrases are selected and evaluated for each single document in the cluster. The candidate words (i.e. nouns and adjectives) of a specified document d in the cluster, which is a subset of V, are marked in the document text, and sequences of adjacent candidate words are collapsed into a multi-word phrase. The phrases ending with an adjective are not allowed, and only the phrases ending with a noun are collected as the candidate phrases for the document. For instance, in the following sentence: "*Mad/JJ cow/NN disease/NN has/VBZ killed/VBN 10,000/CD cattle/NNS"*, the candidate phrases are "*Mad cow disease*" and "*cattle*". The score of a candidate phrase p_i is computed by summing the cluster-level saliency scores of the words contained in the phrase.

$$PhraseScore(p_i) = \sum_{v_j \in p_i} WordScore_{clus}(v_j) \quad (6)$$

All the candidate phrases in the document are ranked in decreasing order of the phrase scores and the top n phrases are selected as the keyphrases of the document. n ranges from 1 to 20 in this study. Similarly for SingleRank, the phrase score is computed based on the document-level saliency scores of the words.

[3] The corresponding POS tags of the candidate words include "JJ", "NN", "NNS", "NNP", "NNPS". We used the Stanford log-linear POS tagger (Toutanova and Manning, 2000) in this study.

4 Empirical Evaluation

4.1 Data Set

To our knowledge, there is no gold standard news dataset with assigned keyphrases for evaluation. So we manually annotated the DUC2001 dataset (Over, 2001) and used the annotated dataset for evaluation in this study. The dataset was originally used for document summarization. It consisted of 309 news articles collected from TREC-9, in which two articles were duplicate (i.e. d05a\FBIS-41815 and d05a\FBIS-41815~). The average length of the documents was 740 words. Two graduate students were employed to manually label the keyphrases for each document. At most 10 keyphrases could be assigned to each document. The annotation process lasted two weeks. The Kappa statistic for measuring inter-agreement among annotators was 0.70. And the annotation conflicts between the two subjects were solved by discussion. Finally, 2488 keyphrases were labeled for the dataset. The average keyphrase number per document was 8.08 and the average word number per keyphrase was 2.09.

The articles have been grouped into 30 clusters manually by NIST annotators for multi-document summarization, and the documents within each cluster were topic-related or relevant. The manually labeled clusters were considered as the ground truth clusters or gold clusters. In order to investigate existing clustering algorithms, the documents in the clusters were mixed together to form the whole document set for automatic clustering.

4.2 Document Clustering Algorithm

In the experiments, several popular clustering algorithms and random clustering algorithms are explored to produce cluster contexts. Note that we have already known the number (i.e. 30) of the clusters for the dataset beforehand and thus we simply use it as input for the following clustering algorithms[4].

Gold Standard Clustering: It is a pseudo clustering algorithm by manually grouping the documents. We use the ground truth clusters as the upperbound of the following automatic clustering algorithms.

Kmeans Clustering: It is a partition based clustering algorithm. The algorithm randomly selects 30 documents as the initial centroids of the 30 clusters and then iteratively assigns all documents to the closest cluster, and recomputes the centroid of each cluster, until the centroids do not change. The similarity between a document and a cluster centroid is computed using the standard Cosine measure.

Agglomerative (AverageLink) Clustering: It is a bottom-up hierarchical clustering algorithm and starts with the points as individual clusters and, at each step, merges the most similar or closest pair of clusters, until the number of the clusters reduces to the desired number 30. The similarity between two clusters is computed using the AverageLink method, which computes the average of the Cosine similarity values between any pair of documents belonging to the two clusters respectively as follows:

$$sim(c_1,c_2) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} sim(d_i,d_j)}{|c_1|\times|c_2|} \quad (7)$$

where d_i, d_j are two documents in cluster c_1 and cluster c_2 respectively, and $|c_1|$ and $|c_2|$ are respectively the numbers of documents in clusters c_1 and c_2.

Agglomerative (CompleteLink) Clustering: It differs from the above agglomerative (AverageLink) clustering algorithm only in that the similarity between two clusters is computed using the CompleteLink method, which computes the minimum of the Cosine similarity values between any pair of documents belonging to the two clusters respectively as follows:

$$sim(c_1,c_2) = \min_{d_i\in c_1, d_j\in c_2}\{sim(d_i,d_j)\} \quad (8)$$

Divisive Clustering: It is a top-down hierarchical clustering algorithm and starts with one, all-inclusive cluster and, at each step, splits the largest cluster (i.e. the cluster with most documents) into two small clusters using the Kmeans algorithm until the number of clusters increases to the desired number 30.

Random Clustering: It produces 30 clusters by randomly assigning each document into one of the k clusters. Three different randomization processes are performed and we denote them as Random1, Random2 and Random3, respectively.

CollabRank relies on the clustering algorithm for document clustering, and the combination of CollabRank and any clustering algorithm will be investigated.

4.3 Evaluation Metric

For evaluation of document clustering results, we adopt the widely used F-Measure to measure the

[4] How to obtain the number of desired clusters is not the focus of this study.

performance of the clustering algorithm (i.e. the quality of the clusters) by comparing the produced clusters with the gold clusters (classes) (Jain et al., 1999).

For evaluation of keyphrase extraction results, the automatic extracted keyphrases are compared with the manually labeled keyphrases. The words are converted to their corresponding basic forms using word stemming before comparison. The precision $p=count_{correct}/count_{system}$, recall $r=count_{correct}/count_{human}$, F-measure ($F=2pr/(p+r)$) are used as evaluation metrics, where $count_{correct}$ is the total number of correct keyphrases extracted by the system, and $count_{system}$ is the total number of automatic extracted keyphrases, and $count_{human}$ is the total number of human-labeled keyphrases.

4.4 Evaluation Results

First of all, we show the document clustering results in Table 1. The gold standard clustering result is the upperbound of all automatic clustering results. Seen from the table, all the four popular clustering algorithms (i.e. CompleteLink, AverageLink, KMeans and Divisive) perform much better than the three random clustering algorithms (i.e. Random1, Random2 and Random3). Different clustering results lead to different document relationships and a high-quality cluster produced by popular algorithms is deemed to build an appropriate cluster context for collaborative keyphrase extraction.

Clustering Algorithm	F-Measure
Gold	1.000
CompleteLink	0.907
AverageLink	0.877
Divisive	0.924
Kmeans	0.866
Random1	0.187
Random2	0.189
Random3	0.183

Table 1. Clustering Results

Now we show the results for keyphrase extraction. In the experiments, the keyphrase number is typically set to 10 and the co-occurrence window size is also simply set to 10. Table 2 gives the comparison results of baseline methods and the proposed CollabRank methods with different clustering algorithms. The TFIDF baseline computes the word scores for each single document based on the word's TFIDF value. The SingleRank baseline computes the word scores for each single document based on the graph-based ranking algorithm. The two baselines do not make use of the cluster context.

Seen from Table 2, the CollabRank methods with the gold standard clustering algorithm or popular clustering algorithms (i.e. Kmeans, CompleteLink, AverageLink and Divisive) perform much better than the baseline methods over all three metrics. The results demonstrate the good effectiveness of the proposed collaborative framework. We can also see that the performance is positively correlated with the clustering results. The CollabRank method with the best performing gold standard clustering results achieves the best performance. While the methods with low-quality clustering results (i.e. the three random clustering results) do not perform well, even much worse than the baseline SingleRank method. This is because that the documents in a low-quality cluster are not truly topic-related, and the mutual influences between the documents are not reliable for evaluating words from a global perspective.

System	Precision	Recall	F-measure
TFIDF	0.232	0.281	0.254
SingleRank	0.247	0.303	0.272
CollabRank (Gold)	**0.283**	**0.348**	**0.312**
CollabRank (Kmeans)	0.276	0.339	0.304
CollabRank (CompleteLink)	0.281	0.345	0.310
CollabRank (AverageLink)	0.277	0.340	0.306
CollabRank (Divisive)	0.274	0.337	0.302
CollabRank (Random1)	0.210	0.258	0.232
CollabRank (Random2)	0.216	0.265	0.238
CollabRank (Random3)	0.209	0.257	0.231

Table 2. Keyphrase Extraction Results

In order to investigate how the co-occurrence window size k and the keyphrase number n influence the performance, we first vary k from 2 to 20 when n is fixed as 10 and the results are shown in Figures 2-4 over three metrics respectively. The results demonstrate that all the methods are not significantly affected by the window size. We then vary n from 1 to 20 when k is fixed as 10 and the results are shown in Figures 5-7. The results demonstrate that the precision values decrease with the increase of n, and the recall values increases with the increase of n, while the F-measure values first increase and then tend to decrease with the increase of n.

We can also see from Figures 2-7 that the CollabRank methods with high-quality clustering results always perform better than the baseline

SingleRank method under different window sizes and different keyphrase numbers, and they always lead to poor performance with low-quality clustering results. This further proves that an appropriate cluster context is very important for the CollabRank method. Fortunately, existing clustering algorithms can obtain the desired cluster context.

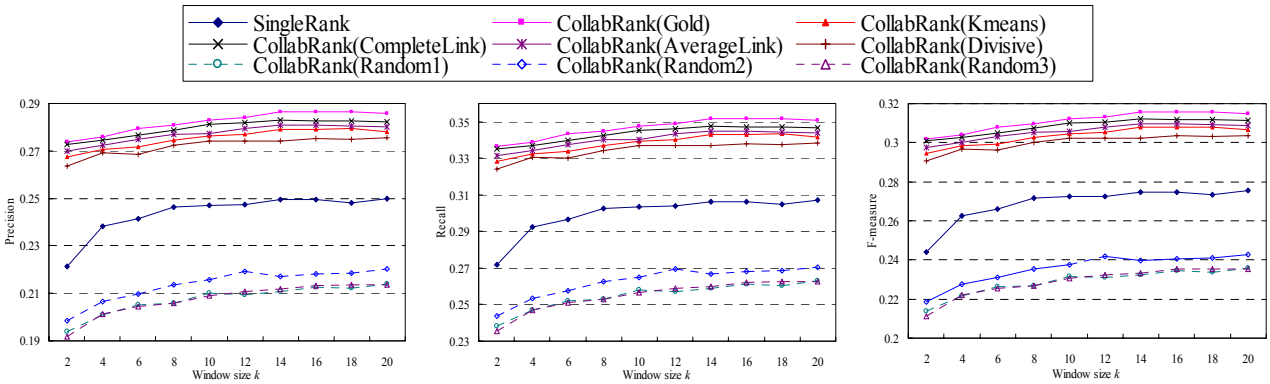

Figure 2. Precision vs. Window Size k

Figure 3. Recall vs. Window Size k

Figure 4. F-measure vs. Window Size k

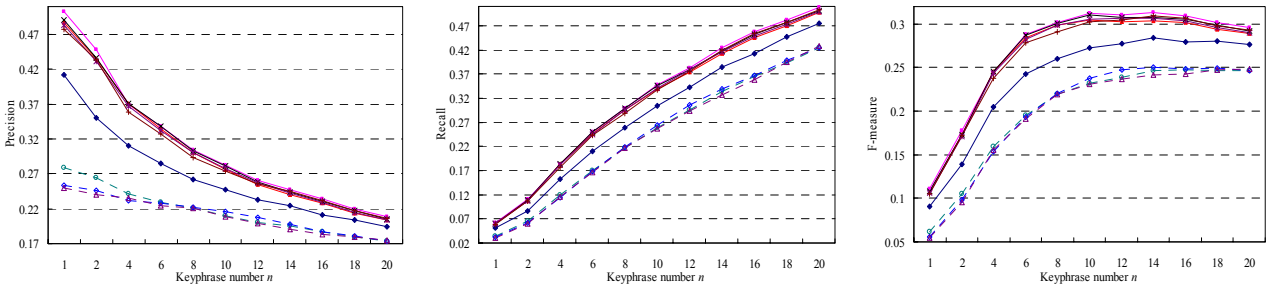

Figure 5. Precision vs. Keyphrase Number n

Figure 6. Recall vs. Keyphrase Number n

Figure 7. F-measure vs. Keyphrase Number n

The proposed CollabRank method makes only use of the global information based on the global graph for the cluster. In order to investigate the relative contributions from the whole cluster and the single document to the final performance, we experiment with the method named RankFusion which makes both of the cluster-level global information and the document-level local information. The overall word score $WordScore_{fusion}(v_i)$ for word v_i in a document in RankFusion is a linear combination of the global word score and the local word score as follows:

$$WordScore_{fusion}(v_i) = \lambda \cdot WordScore_{clus}(v_i) + (1-\lambda) \cdot WordScore_{doc}(v_i) \quad (9)$$

where $\lambda \in [0,1]$ is the fusion weight. Then the phrase score is computed based on the fusion scores of the words. The RankFusion method is the same with CollabRank if $\lambda=1$ and it is the same with SingleRank if $\lambda=0$.

Figure 8 shows the F-measure curves for the RankFusion methods with different high-quality clustering algorithms under different fusion weights. We can see that when $\lambda \in (0.5,1)$, the RankFusion methods with high-quality clusters can outperform both the corresponding SingleRank and the corresponding CollabRank. However, the performance improvements of RankFusion over CollabRank are not significant. We can conclude that the cluster-level global information plays the key role for evaluating the true saliency of the words.

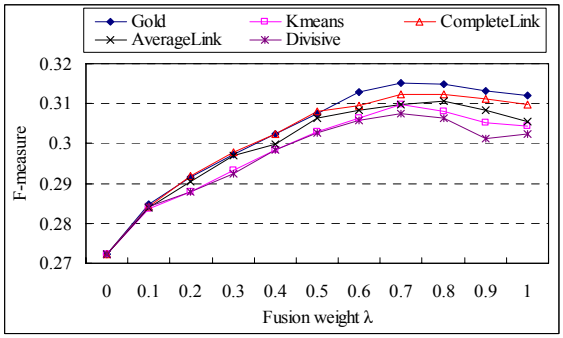

Figure 8. RankFusion Results (F-measure) vs. Fusion Weight λ

5 Conclusion and Future Work

In this paper, we propose a novel approach named CollabRank for collaborative single-document keyphrase extraction, which makes use of the mutual influences between documents in appropriate cluster context to better evaluate the saliency of words and phrases. Experimental re-

sults demonstrate the good effectiveness of CollabRank. We also find that the clustering algorithm is important for obtaining the appropriate cluster context and the low-quality clustering results will deteriorate the extraction performance. It is encouraging that most existing popular clustering algorithms can meet the demands of the proposed approach.

The proposed collaborative framework has more implementations than the implementation based on the graph-based ranking algorithm in this study. In future work, we will explore other keyphrase extraction methods in the proposed collaborative framework to validate the robustness of the framework.

Acknowledgements

This work was supported by the National Science Foundation of China (No.60703064), the Research Fund for the Doctoral Program of Higher Education of China (No.20070001059) and the National High Technology Research and Development Program of China (No.2008AA01Z421).

References

A. Berger and V. Mittal. 2000. OCELOT: A system for summarizing Web Pages. In *Proceedings of SIGIR2000*.

K. Barker and N. Cornacchia. 2000. Using nounphrase heads to extract document keyphrases. In *Canadian Conference on AI*.

O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. 2001. Seeing the whole in parts: text summarization for web browsing on handheld devices. In *Proceedings of WWW2001*.

M. Chen, J.-T. Sun, H.-J. Zeng and K.-Y. Lam. 2005. A practical system for keyphrase extraction for web pages. In *Proceedings of CIKM2005*.

E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. *Proceedings of IJCAI-99*, pp. 668-673.

C. Gutwin, G. W. Paynter, I. H. Witten, C. G. Nevill-Manning and E. Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Journal of Decision Support Systems*, 27, 81-104.

K. M. Hammouda, D. N. Matute and M. S. Kamel. 2005. CorePhrase: keyphrase extraction for document clustering. In *Proceedings of MLDM2005*.

A. Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of EMNLP2003*, Japan, August.

A. K. Jain, M. N. Murty and P. J. Flynn. 1999. Data clustering: a review. *ACM Computing Surveys*, 31(3):264-323.

D. Kelleher and S. Luz. 2005. Automatic hypertext keyphrase detection. In *Proceedings of IJCAI2005*.

B. Krulwich and C. Burkey. 1996. Learning user information interests through the extraction of semantically significant phrases. In *AAAI 1996 Spring Symposium on Machine Learning in Information Access*.

O. Medelyan and I. H. Witten. 2006. Thesaurus based automatic keyphrase indexing. In *Proceedings of JCDL2006*.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP2004*.

A. Muñoz. 1996. Compound key word generation from document databases using a hierarchical clustering ART model. *Intelligent Data Analysis,* 1(1).

T. D. Nguyen and M.-Y. Kan. 2007. Keyphrase extraction in scientific publications. In *Proceedings of ICADL2007*.

P. Over. 2001. Introduction to DUC-2001: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of DUC2001*.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. *Technical report*, Stanford Digital Libraries.

M. Song, I.-Y. Song and X. Hu. 2003. KPSpotter: a flexible information gain-based keyphrase extraction system. In *Proceedings of WIDM2003*.

A. M. Steier and R. K. Belew. 1993. Exporting phrases: A statistical analysis of topical language. In *Proceedings of Second Symposium on Document Analysis and Information Retrieval*, pp. 179-190.

T. Tomokiyo and M. Hurst. 2003. A language model approach to keyphrase extraction. In: *Proceedings of ACL Workshop on Multiword Expressions*.

K. Toutanova and C. D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy Part-of-Speech tagger. In *Proceedings of EMNLP/VLC-2000*.

P. D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303-336.

P. D. Turney. 2003. Coherent keyphrase extraction via web mining. In *Proc. of IJCAI-03*, pages 434–439.

X. Wan, J. Yang and J. Xiao. 2007a. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of ACL2007*.

I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. *Proceedings of Digital Libraries 99 (DL'99),* pp. 254-256.

W.-T. Yih, J. Goodman and V. R. Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of WWW2006*.

H. Y. Zha. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of SIGIR2002*, pp. 113-120.

Y. Zhang, N. Zincir-Heywood, and E. Milios. 2004. Term-Based Clustering and Summarization of Web Page Collections. In *Proceedings of the Seventeenth Conference of the Canadian Society for Computational Studies of Intelligence*.

Y. Zhang, N. Zincir-Heywood and E. Milios. 2005. Narrative text classification for automatic key phrase extraction in web document corpora. In *Proceedings of WIDM2005*.