WordTopic-MultiRank : A New Method for Automatic Keyphrase Extraction

Fan Zhang[†] **Lian'en Huang**[†] **Bo Peng**[‡]
[†]The Shenzhen Key Lab for Cloud Computing Technology and Applications
Peking University Shenzhen Graduate School, Shenzhen 518055, P.R.China
fan.zhgf@gmail.com, hle@net.pku.edu.cn
[‡]Institute of Network Computing and Information Systems
Peking University, Beijing 100871, P.R.China
pb@net.pku.edu.cn

Abstract

Automatic keyphrase extraction aims to pick out a set of terms as a representation of a document without manual assignment efforts. Supervised and unsupervised graph-based ranking methods have been studied for this task. However, previous methods usually computed importance scores of words under the assumption of single relation between words. In this work, we propose WordTopic-MultiRank as a new method for keyphrase extraction, based on the idea that words relate with each other via multiple relations. First we treat various latent topics in documents as heterogeneous relations between words and construct a multi-relational word network. Then, a novel ranking algorithm, named Biased-MultiRank, is applied to score the importance of words and topics simultaneously, as words and topics are considered to have mutual influence on each other. Experimental results on two different data sets show the outstanding performance and robustness of our proposed approach in automatic keyphrase extraction task.

1 Introduction

Keyphrases refer to the meaningful words and phrases that can precisely and compactly represent documents. Appropriate keyphrases help users a lot in better grasping and remembering key ideas of articles, as well as fast browsing and reading. Moreover, qualities of some information retrieval and natural language processing tasks have been improved with the help of document keyphrases, such as document indexing, categorizing, clustering and summarizing (Gutwin et al., 1999; Krulwich and Burkey, 1996; Hammouda et al., 2005).

Usually, keyphrases are manually assigned by authors, which is time consuming. With the fast development of Internet, it becomes impractical to label them by human effort as articles on the Web increase exponentially. Therefore, automatic keyphrase extraction plays an important role in keyphrases assignment task.

In most existing work, words are assumed under a single relation and then scored or judged within it. Considering the famous TextRank (Mihalcea and Tarau, 2004), a term graph under a single relatedness was built first, then a graph-based ranking algorithm, such as PageRank (Page et al., 1999), was used to determine the importance score for each term. Another compelling example is (Liu et al., 2010), where words were scored under each topic separately.

In this study, inspired by some multi-relational data mining techniques, such as (Ng et al., 2011), we assume each topic as a single relation type and construct an intra-topic word network for each relation type. In other words, it is to map word relatedness within multiple topics to heterogeneous relations, meaning that words have interactions with others based on different topics.

A multi-relational words example of our proposed WordTopic-MultiRank model is shown in Figure 1(a). There are four words and three relations in this example, implying that there are three potential topics contained in the document. Further, we represent such multi-relational data in a tensor shape in Figure 1(b), where each two-dimensional plane represents an adjacency matrix for one type of topics. Then the heterogeneous network can be depicted as a tensor of size $4 \times 4 \times 3$, where (i, j, k) entry is nonzero if the ith word is related to the jth word under kth topic.

After that, we raise a novel measurement of word relatedness considering different topics, and

[‡]Corresponding author.

International Joint Conference on Natural Language Processing, pages 10–18,
Nagoya, Japan, 14-18 October 2013.

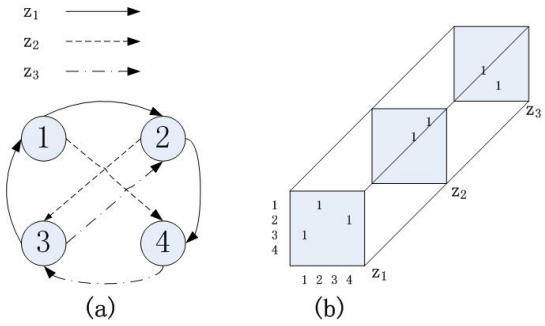

Figure 1: (a) An example of multi-relational words in graph representation and (b) the corresponding tensor representation.

then apply Biased-MultiRank algorithm to deal with multi-relational words for co-ranking purpose, based on the idea that words and topics have mutual influence on each other. More specifically, a word, connected with highly scored words via highly scored topics, should receive a high score itself, and similarly, a topic, connecting highly scored words, should get a high score as well.

Experiments have been performed on two different data sets. One is a collection of scientific publication abstracts, while the other consists of news articles with human-annotated keyphrases. Experimental results demonstrate that our WordTopic-MultiRank method outperforms representative baseline approaches in specified evaluation metrics. And we have investigated how different parameter values influence the performance of our method.

The rest of this paper is organized as follows. Section 2 introduces related work. In Section 3, details of constructing and applying WordTopic-MultiRank model are presented. Section 4 shows experiments and results on two different data sets. Finally, in Section 5, conclusion and future work are discussed .

2 Related Work

Existing methods for keyphrase extraction task can be divided into supervised and unsupervised approaches. The supervised methods mainly treat keyphrase extraction as a classification task, so a model needs to be trained before classifying whether a candidate phrase is a keyphrase or not. Turney (1999) firstly utilized a genetic algorithm with parameterized heuristic rules for keyphrase extraction, then Hulth (2003) added more linguistic knowledge as features to achieve better performance. Jiang et al. (2009) employed linear Ranking SVM, a learning to rank method, to extract keyphrase lately. However, supervised methods require a training set which would demand time-consuming human-assigned work, making it impractical in the vast Internet space. In this work, we principally concentrate on unsupervised methods.

Among those unsupervised approaches, clustering and graph-based ranking methods showed good performance in this task. Representative studies of clustering approaches are (Liu et al., 2009) and (Grineva et al., 2009). Liu et al. (2009) made use of clustering methods to find exemplar terms and then selected terms from each cluster as keyphrases. Grineva et al. (2009) applied graph community detection techniques to partition the term graph into thematically cohesive groups and selected groups that contained key terms, discarding groups with unimportant terms. But as is widely known, one of the major difficulties in clustering is to predefine the cluster number which influences performance heavily.

As for basic graph-based approaches, such as (Mihalcea and Tarau, 2004) and (Litvak and Last, 2008), a graph based on word linkage or word similarity was first constructed, then a ranking algorithm was used to determine the importance score of each term. Wan et al. (2007) presented an idea of extracting summary and keywords simultaneously under the assumption that summary and keywords of the same document can be mutually boosted. Moreover, Wan and Xiao (2008a) used a small number of nearest neighbor documents for providing more knowledge to improve performance and similarly, Wan and Xiao (2008b) made use of multiple documents with a cluster context. Recently, topical information was under consideration to be combined with graph-based approaches. One of the outstanding studies was Topic-sensitive PageRank (Haveliwala, 2002), which computed scores of web pages by incorporating topics of the context. As another representative, Topical PageRank (Liu et al., 2010) applied a Biased PageRank to assign an importance score to each term under every latent topic separately.

To the best of our knowledge, previous graph-based researches are based on the assumption that all words exist under a unified relation, while in this work, we view latent topics within documents

as word relations and words as multi-relational data, in order to make full use of word-word relatedness, word-topic interaction and inter-topic impacts.

3 WordTopic-MultiRank Method

In this section, we will introduce our proposed WordTopic-MultiRank method in details, including topic decomposition, word relatedness measurement, heterogeneous network construction and Biased-MultiRank algorithm.

3.1 Topic Detection via Latent Dirichlet Allocation

There are some existing methods to infer latent topics of words and documents. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is adopted in our work as it is more feasible for inference and it can reduce the risk of over-fitting.

Firstly, we denote the learning corpus for LDA as C, and $|C|$ represents the total number of documents in C. The i_{th} document in the corpus is denoted as d_i, in which $i = 1, 2, \cdots, |C|$. Then, words are denoted as w_{ij} where i indicates that word w_{ij} appears in document d_i and j refers to jth position in d_i ($j = 1, 2, \cdots, |d_i|$, $|d_i|$ is the total word number in d_i). Further, topics inferred from $|C|$ is z_k, $k = 1, 2, \cdots, |T|$, while T stands for the topic set detected from C and $|T|$ is the total number of topics.

According to LDA, observed words in each document are supposed to be generated by a document-specific mixture of corpus-wide latent topics. More specifically, each word w_{ij} in document d_i is generated by first sampling a topic z_k from d_i's document-topic multinomial distribution θ_{d_i}, and then sampling a word from z_k's topic-word multinomial distribution ϕ_{z_k}. And each θ_{d_i} is generated by a conjugate Dirichlet prior with parameter α, while each ϕ_{z_k} is generated by a conjugate Dirichlet prior with parameter β. The full generative model for w_{ij} is given by:

$$p(w_{ij}|d_i, \alpha, \beta) = \sum_{k=1}^{|T|} p(w_{ij}|z_k, \beta)p(z_k|d_i, \alpha) \quad (1)$$

Using LDA, we finally obtain the document-topic distribution, namely $p(z_k|d_i)$ for all the topics z_k on each document d_i, as well as the topic-word distribution, namely $p(w_{ij}|z_k)$ for all the words w_{ij} on each topic z_k.

In this work, we use GibbsLDA++[1], a C/C++ implementation of LDA using Gibbs Sampling, to detect latent topics.

3.2 Measurement of Word Relatedness under Multi-relations

Next, we apply Bayes' theorem to get word-topic distribution $p(z_k|w_{ij})$ for every word in a given document d_i:

$$p(z_k|w_{ij}) = \frac{p(w_{ij}|z_k, \beta)p(z_k|d_i, \alpha)}{\sum_{k=1}^{|T|} p(w_{ij}|z_k, \beta)p(z_k|d_i, \alpha)} \quad (2)$$

Therefore, we can obtain word relatedness as follows:

$$p(w_{im}|w_{in}, z_k) = p(w_{im}|z_k)p(z_k|w_{in}) \quad (3)$$

where $m, n = 1, 2, \cdots, |d_i|$, and $p(w_{im}|w_{in}, z_k)$ represents the relatedness of word w_{im} and word w_{in} under kth topic.

From the view of probability, $p(z_k|w_{in})$ is the probability of word w_{in} being assigned to topic z_k and $p(w_{im}|z_k)$ is the probability of generating word w_{im} from the same topic z_k. Therefore, $p(w_{im}|w_{in}, z_k)$ shows the probability of generating word w_{im} if we have observed word w_{in} under topic z_k. Obviously, this point of view corresponds with LDA and it connects words via topics.

3.3 Constructing a Heterogeneous Network on Words

Like Figure 1(a) shown in Introduction, now we construct a multi-relational network for words. In the same way mentioned by typical graph-based methods, for every document d_i in corpus C, we treat every single word as a vertex and make use of word co-occurrences to construct a word graph as it indicates the cohesion relationship between words in the context of document d_i. In this process, a sliding window with maximum W words is used upon the word sequences of documents. Those words appearing in the same window will have a link to each other under all the relations in the network.

Further, we obtain the word relatedness under every topic from Formula (3), and use them as weights of edges for constructing the heterogeneous network. For instance, $p(w_{im}|w_{in}, z_k)$ is regarded as the weight of the edge from w_{in} to w_{im} under kth relation if there is a co-occurrence relation between the two words in document d_i.

[1] GibbsLDA++: http://gibbslda.sourceforge.net

As (Hulth, 2003) pointed out, most manually assigned keyphrases were noun groups whose pattern was zero or more adjectives followed by one or more nouns. We only take adjectives and nouns into consideration while constructing networks in experiments.

3.4 Ranking Algorithm

In our proposed method, we employ Biased-MultiRank algorithm for co-ranking the importance of words and topics. It is obtained by adding prior knowledge of words and topics to Basic-MultiRank, a basic co-ranking scheme designed for objects and relations in multi-relational data. Therefore, we will demonstrate Basic-MultiRank first, then derive Biased-MultiRank algorithm from it.

3.4.1 Basic-MultiRank Algorithm

In this subsection, we take document d_i into discussion for convenience. First, we call $\mathcal{A} = (a_{w_{im},w_{in},z_k})$ a real $(2,1)$th order $(|d_i| \times |T|)$-dimensional rectangular tensor, where a_{w_{im},w_{in},z_k} denotes $p(w_{im}|w_{in},z_k)$ obtained in last subsection, in which $m,n = 1,2,\cdots,|d_i|$ and $k = 1,2,\cdots,|T|$. For example, Figure 1(b) is a $(2,1)$th order (4×3)-dimensional tensor representation of a document, in which there are 4 words and 3 topics.

Then two transition probability tensors $\mathcal{O} = (o_{w_{im},w_{in},z_k})$ and $\mathcal{R} = (r_{w_{im},w_{in},z_k})$ are constructed with respect to words and topics by normalizing all the entries of $\mathcal{A}$:

$$o_{w_{im},w_{in},z_k} = \frac{a_{w_{im},w_{in},z_k}}{\sum_{m=1}^{|d_i|} a_{w_{im},w_{in},z_k}} \tag{4}$$

$$r_{w_{im},w_{in},z_k} = \frac{a_{w_{im},w_{in},z_k}}{\sum_{k=1}^{|T|} a_{w_{im},w_{in},z_k}} \tag{5}$$

Here we deal with dangling node problem in the same way as PageRank (Page et al., 1999). Namely, if a_{w_{im},w_{in},z_k} is equal to 0 for all words w_{im}, which means that word w_{in} had no link out to any other words via topic z_k, we set o_{w_{im},w_{in},z_k} to be $1/|d_i|$. Likewise, if a_{w_{im},w_{in},z_k} is equal to 0 for all z_k, which means that word w_{in} had no link out to words w_{im} via all topics, we set r_{w_{im},w_{in},z_k} to be $1/|T|$. In this way, we ensure that

$$0 \le o_{w_{im},w_{in},z_k} \le 1, \sum_{m=1}^{|d_i|} o_{w_{im},w_{in},z_k} = 1$$

$$0 \le r_{w_{im},w_{in},z_k} \le 1, \sum_{k=1}^{|T|} r_{w_{im},w_{in},z_k} = 1$$

Following the rule of Markov chain, we derive the probabilities like:

$$P[X_t{=}w_{im}]{=}\sum_{n=1}^{|d_i|}\sum_{k=1}^{|T|} o_{w_{im},w_{in},z_k} \times P[X_{t-1}{=}w_{in},Y_t{=}z_k] \tag{6}$$

$$P[Y_t{=}z_k]{=}\sum_{m=1}^{|d_i|}\sum_{n=1}^{|d_i|} r_{w_{im},w_{in},z_k} \times P[X_t{=}w_{im},X_{t-1}{=}w_{in}] \tag{7}$$

where subscript t denotes the iteration number.

Notice that Formula (6) and (7) accord with our basic idea that, a word connected with high probability words via high probability relations, should have a high probability so that it will be visited more likely, and a topic connecting words with high probabilities, should also get a high one.

After employing a product form of individual probability distributions, we decouple the two joint probability distributions in Formula (6) and (7) as follows:

$$P[X_{t-1}{=}w_{in},Y_t{=}z_k]{=}P[X_{t-1}{=}w_{in}]P[Y_t{=}z_k] \tag{8}$$

$$P[X_t{=}w_{im},X_{t-1}{=}w_{in}]{=}P[X_t{=}w_{im}]P[X_{t-1}{=}w_{in}] \tag{9}$$

Considering stationary distributions of words and topics, while t goes infinity, the WordTopic-MultiRank values are given by:

$$\overline{\mathbf{x}}{=}[\overline{x}_{w_{i1}},\overline{x}_{w_{i2}},\cdots,\overline{x}_{w_{i|d_i|}}]^T \tag{10}$$

$$\overline{\mathbf{y}}{=}[\overline{y}_{z_1},\overline{y}_{z_2},\cdots,\overline{y}_{z_{|T|}}]^T \tag{11}$$

with

$$\overline{x}_{w_{im}}{=}\lim_{t\to\infty} P[X_t{=}w_{im}] \tag{12}$$

$$\overline{y}_{z_k}{=}\lim_{t\to\infty} P[Y_t{=}z_k] \tag{13}$$

Under the assumptions from Formula (8) to (13), we can derive these from Formula (6) and (7):

$$\overline{x}_{w_{im}}{=}\sum_{n=1}^{|d_i|}\sum_{k=1}^{|T|} o_{w_{im},w_{in},z_k}\overline{x}_{w_{in}}\overline{y}_{z_k} \tag{14}$$

$$\overline{y}_{z_k}{=}\sum_{m=1}^{|d_i|}\sum_{n=1}^{|d_i|} r_{w_{im},w_{in},z_k}\overline{x}_{w_{im}}\overline{x}_{w_{in}} \tag{15}$$

which mean that the score of w_{im} depends on its weighted-links with other words via all topics and the score of z_k depends on scores of the words which it connects with.

Now we are able to solve two tensor equations shown below to obtain the WordTopic-MultiRank values of words and relations according to tensor operations Formula (14) and (15):

$$\mathcal{O}\overline{\mathbf{x}}\overline{\mathbf{y}}=\overline{\mathbf{x}} \quad (16)$$

$$\mathcal{R}\overline{\mathbf{x}}^2=\overline{\mathbf{y}} \quad (17)$$

Ng et al. (2011) show the existence and uniqueness of stationary probability distributions $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$, then propose MultiRank, an iterative algorithm, to solve Formula (16) and (17) utilizing Formula (14) and (15). We refer it as Basic-MultiRank algorithm, shown as **Algorithm 1**, for the reason that it will be modified later in the following subsection.

Algorithm 1 Basic-MultiRank algorithm

Require: Tensor $\mathcal{A}$, initial probability distributions $\overline{\mathbf{x}}_0$ and $\overline{\mathbf{y}}_0$ ($\sum_{m=1}^{|d_i|}[\overline{\mathbf{x}}_0]_{w_m}=1$ and $\sum_{k=1}^{|T|}[\overline{\mathbf{y}}_0]_{z_k}=1$) , tolerance ϵ

Ensure: Two stationary probability distributions $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$

1: compute tensor $\mathcal{O}$ and $\mathcal{R}$;
2: set t = 1;
3: Compute $\overline{\mathbf{x}}_t=\mathcal{O}\overline{\mathbf{x}}_{t-1}\overline{\mathbf{y}}_{t-1}$;
4: Compute $\overline{\mathbf{y}}_t=\mathcal{R}\overline{\mathbf{x}}_t^2$;
5: if $||\overline{\mathbf{x}}_t-\overline{\mathbf{x}}_{t-1}||+||\overline{\mathbf{y}}_t-\overline{\mathbf{y}}_{t-1}||<\epsilon$, then stop, otherwise set $t=t+1$ and goto Step 3;
6: **return** $\overline{\mathbf{x}}_t$ and $\overline{\mathbf{y}}_t$.

3.4.2 Biased-MultiRank Algorithm

Inspired by the idea of Biased PageRank (Liu et al., 2010), we treat document-word distribution $p(w_{ij}|d_i)$, which can be computed from Formula (1), and document-topic distribution $p(z_k|d_i)$, acquired from topic decomposition, as prior knowledge for words and topics in each document d_i. Therefore, we modify Formula (16) and (17) by adding prior knowledge to it as follows:

$$(1-\lambda)\mathcal{O}\overline{\mathbf{x}}\overline{\mathbf{y}}+\lambda\overline{\mathbf{x}}_p=\overline{\mathbf{x}} \quad (18)$$

$$(1-\gamma)\mathcal{R}\overline{\mathbf{x}}^2+\gamma\overline{\mathbf{y}}_p=\overline{\mathbf{y}} \quad (19)$$

where, $\overline{\mathbf{x}}_p=[p(w_{i1}|d_i),p(w_{i2}|d_i),\cdots,p(w_{i|d_i|}|d_i)]^T$ and $\overline{\mathbf{y}}_p=[p(z_1|d_i),p(z_2|d_i),\cdots,p(z_{|T|}|d_i)]^T$.

Then we propose Biased-MultiRank, shown as **Algorithm 2**, as a new algorithm to solve the prior-tensors and Formula (18) and (19). Finally it is used in our WordTopic-MultiRank model.

Algorithm 2 Biased-MultiRank algorithm

Require: Tensor $\mathcal{A}$, initial probability distributions $\overline{\mathbf{x}}_0$ and $\overline{\mathbf{y}}_0$ ($\sum_{m=1}^{|d_i|}[\overline{\mathbf{x}}_0]_{w_m}=1$ and $\sum_{k=1}^{|T|}[\overline{\mathbf{y}}_0]_{z_k}=1$), prior distribution of words $\overline{\mathbf{x}}_p$ and topics $\overline{\mathbf{y}}_p$, parameters λ and γ ($0\leq\lambda,\gamma<1$), tolerance ϵ

Ensure: Two stationary probability distributions $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$

1: compute tensors $\mathcal{O}$ and $\mathcal{R}$;
2: set t = 1;
3: Compute $\overline{\mathbf{x}}_t=(1-\lambda)\mathcal{O}\overline{\mathbf{x}}_{t-1}\overline{\mathbf{y}}_{t-1}+\lambda\overline{\mathbf{x}}_p$;
4: Compute $\overline{\mathbf{y}}_t=(1-\gamma)\mathcal{R}\overline{\mathbf{x}}_t^2+\gamma\overline{\mathbf{y}}_p$;
5: if $||\overline{\mathbf{x}}_t-\overline{\mathbf{x}}_{t-1}||+||\overline{\mathbf{y}}_t-\overline{\mathbf{y}}_{t-1}||<\epsilon$, then stop, otherwise set $t=t+1$ and goto Step 3;
6: **return** $\overline{\mathbf{x}}_t$ and $\overline{\mathbf{y}}_t$.

4 Experiment

To evaluate the performance of WordTopic-MultiRank in automatic keyphrase extraction task, we utilize it on two different data sets and describe the experiments specifically in this section.

4.1 Experiments on Scientific Abstracts

4.1.1 Data Set

We first employ WordTopic-MultiRank model to conduct experiments on a data set of scientific publication abstracts from the INSPEC database with corresponding manually assigned keyphrases[2]. The data set is also used by Hulth (2003), Mihalcea and Tarau (2004), Liu et al. (2009), and Liu et al. (2010), meaning that it is classically used in the task of keyphrase extraction, and is convenient for comparison.

Actually, this data set contains 2,000 abstracts of research articles and 19,254 manually annotated keyphrases, and is split into 1,000 for training, 500 for validation and 500 for testing.

In this study, we use the 1,000 training documents as corpus C for topic detection and like other unsupervised ranking methods, 500 test documents are used for comparing the performance with baselines. Following previous work, only the manually assigned uncontrolled keyphrases that occur in the corresponding abstracts are viewed as standard answers.

[2]It can be obtained from https://github.com/snkim/AutomaticKeyphraseExtraction

4.1.2 Baselines and Evaluation Metrics

We choose methods proposed by Hulth (2003), Mihalcea and Tarau (2004), Liu et al. (2009), and Liu et al. (2010) as baselines for the reason that they are either classical or outstanding in keyphrase extraction task.

Evaluation metrics are *precision, recall, F1-measure* shown as follows:

$$P=\frac{TP}{TP+FP},\ R=\frac{TP}{TP+FN},\ F1=\frac{2PR}{P+R} \quad (20)$$

where TP is the total number of correctly extracted keyphrases, FP is the number of incorrectly extracted keyphrases, and FN is the number of those keyphrases which are not extracted.

4.1.3 Data Pre-processing and Configuration

Documents are pre-processed by removing stop words and annotated with POS tags using Stanford Log-Linear Tagger[3].

Based on the research result of (Hulth, 2003), only adjectives and nouns are used in constructing multi-relational words network for ranking, and keyphrases corresponding with following pattern are considered as candidates:

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

in which, JJ indicates adjectives while NN, NNS and NNP represent various forms of nouns.

At last, top-M keyphrases, which have highest sum scores of words contained in them, are extracted and compared with standard answers after stemming by Porter stemmer[4].

In experiments, we set α=1, β=0.01 for Formula (1) to (3) empirically, and λ=0.5, γ=0.9 for Formula (18), (19) indicated by (Li et al., 2012). Influences of these parameters will not be discussed further in this work as they have been studied intensively in previous researches.

4.1.4 Experimental Results

In this subsection, we investigate how different parameter values influence performance of our proposed model first, then compare the best results obtained by baseline methods and our model.

First of all, we inspect influences of topic number $|T|$ on our model performance. Table 1 shows experimental results when $|T|$ ranges from 20 to 100 while setting window size W=2 and max extracted number M=10.

[3]http://nlp.stanford.edu/software/tagger.shtml

[4]http://tartarus.org/martin/PorterStemmer/

Topic Number	Precision	Recall	F1
20	0.463	0.498	0.479
40	0.464	0.500	0.480
60	0.465	0.502	**0.482**
80	0.462	0.499	0.480
100	0.462	0.499	0.480

Table 1: Influence of Topic Number $|T|$

From Table 1, we observe that the performance does not change much when the number of topics varies, showing our model's robustness under the situation that the actual number of topics is unknown, which is commonly seen in Information Retrieval and Natural Language Processing applications. We can see that $|T|$=60 produces the best result for this corpus, so we choose 60 for $|T|$ in comparison with baselines.

Then, we fix $|T|$=60 and M=10 to demonstrate how our model is affected by the windows size W. Table 2 presents the metrics when W ranges from 2 to 10.

Window Size	Precision	Recall	F1
2	0.465	0.502	**0.482**
4	0.461	0.496	0.477
6	0.462	0.500	0.480
8	0.461	0.499	0.479
10	0.461	0.498	0.478

Table 2: Influence of Window Size W

Our results are consistent with the findings reported by Liu et al. (2009) and Liu et al. (2010), indicating that performance usually does not vary much when W ranges. More details point out that W=2 is the best.

Moreover, we explore the influence of max extracted number M by setting W=2 and $|T|$=60.

M	Precision	Recall	F1
5	**0.602**	0.393	0.475
10	0.465	0.502	**0.482**
15	0.420	**0.550**	0.476

Table 3: Influence of Max Extracted Number M

Table 3 indicates that as M increases, *precision* falls down while *recall* raises up, and M=10 performs best in *F1-measure*.

At last, Table 4 shows the best results of baseline methods and our proposed model. In fac-

Method	Precision	Recall	F1
Hulth's (Hulth, 2003)	0.252	0.517	0.339
TextRank (Mihalcea and Tarau, 2004)	0.312	0.431	0.362
Topical PageRank (Liu et al., 2010)	0.354	0.183	0.242
Clustering (Liu et al., 2009)	0.350	**0.660**	0.457
WordTopic-MultiRank	**0.465**	0.502	**0.482**

Table 4: Comparison on Scientific Abstracts

Method	Precision	Recall	F1
ExpandRank(Wan and Xiao, 2008a)	0.288	0.354	0.317
CollaRank(Wan and Xiao, 2008b)	0.283	0.348	0.312
Topical PageRank(Liu et al., 2010)	0.282	0.348	0.312
WordTopic-MultiRank	**0.296**	**0.399**	**0.340**

Table 5: Comparison on DUC2001

t, the best result of (Hulth, 2003) was obtained by adding POS tags as features for classification, while running PageRank on an undirected graph, which was built via using window W=2 on word sequence, resulted best of (Mihalcea and Tarau, 2004). According to (Liu et al., 2009), spectral clustering method got best performance in *precision* and *F1-measure*. On the other hand, Topical PageRank (Liu et al., 2010) performed best when setting window size W=10, topic number $|T|$=1,000. Since the influences of parameters have been discussed above, we set W=2, $|T|$=60 and M=10 as they result in best performance of our model on the same data set.

Table 4 demonstrates that our proposed model outperforms all baselines in both *precision* and *F1-measure*. Noting that baseline methods are all under a single relation type assumption for word relatedness, estimations of their word ranking scores are limited, while WordTopic-MultiRank assumes words as multi-relational data and considers interactions between words and topics more comprehensively.

4.2 Experiments on DUC2001

In order to show the generalization performance of our model, we also conduct experiments on another data set for automatic keyphrase extraction task and describe it in this subsection briefly.

Following (Wan and Xiao, 2008a), (Wan and Xiao, 2008b) and (Liu et al., 2010), a data set annotated by Wan and Xiao[5] was used in this experiment for evaluation. This data set is the testing part of DUC2001(Over and Yen, 2004), containing 308 news articles with 2,488 keyphrases manually labeled. And at most 10 keyphrases were assigned to each document. Again, we choose *precision*, *recall* and *F1-measure* as evaluation metrics and use the train part of DUC2001 for topic detection. At last, keyphrases extracted by our WordTopic-MultiRank model will be compared with the ones occurring in corresponding articles after stemming.

As indicated in (Wan and Xiao, 2008b), performance on test set does not change much when co-occurrence window size W ranges from 5 to 20, and (Liu et al., 2010) also reports that it does not change much when topic number ranges from 50 to 1,500. Therefore, we pick co-occurrence window size W=10 and topic number $|T|$=60 to run WordTopic-MultiRank model. As for Keyphrase number M, we vary it from 1 to 20 to obtain different performances. Results are shown in Figure 2.

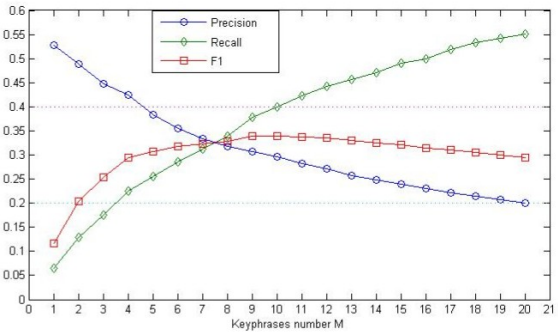

Figure 2: performance vs. Keyphrase number M

From Figure 2, we can observe how performances of our model change with M. Actually,

[5]http://wanxiaojun1979.googlepages.com/

as M increases from 1 to 20, *precision* decreases from 0.528 to 0.201 in our experiment, while *recall* increases from 0.065 to 0.551. As for *F1-measure*, it obtains maximum value 0.340 when M=10 and decreases gradually as M leaves 10 farther. Therefore, W=10, $|T|$=60 and M=10 are optimal for our proposed method on this test set.

Table 5 lists the best performance comparison between our method and previous ones. All previous methods perform best on DUC2001 test set while setting co-occurrence window size W=10 and Keyphrase number M=10, which is consistent with our model.

Experimental results on this data set demonstrate the effectiveness of our proposed model again as it outperforms baseline methods over all three metrics.

5 Conclusion and Future Work

In this study, we propose a new method named WordTopic-MultiRank for automatic keyphrase extraction task. It treats words in documents as objects and latent topics as relations, assuming words are under multiple relations. Based on the idea that words and topics have mutual influence on each other, our model ranks importance of words and topics simultaneously, then extracts highly scored phrases as keyphrases. In this way, it makes full use of word-word relatedness, word-topic interaction and inter-topic impacts. Experiments demonstrate that WordTopic-MultiRank achieves better performance than baseline methods on two different data sets. It also shows the good effectiveness and strong robustness of our method after we explored the influence of different parameter values.

In future work, for one thing, we would like to investigate how different corpora influence our method and choose a large-scale and general corpus, such as Wikipedia, for experiments. For another, exploring more algorithms to deal with heterogeneous relation network may help to unearth more knowledge between words and topics, and improve our model performance.

Acknowledgments

This research is financially supported by NSFC Grant 61073082 and NSFC Grant 61272340.

References

D.M. Blei, A.Y. Ng, and M.I. Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th international conference on World wide web*, pages 661–670. ACM.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1):81–104.

Khaled M Hammouda, Diego N Matute, and Mohamed S Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *Machine Learning and Data Mining in Pattern Recognition*, pages 265–274. Springer.

Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.

A. Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of EMNLP*, pages 216–223.

Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 756–757. ACM.

Bruce Krulwich and Chad Burkey. 1996. Learning user information interests through extraction of semantically significant phrases. In *Proceedings of the AAAI spring symposium on machine learning in information access*, pages 100–112.

Xutao Li, Michael K Ng, and Yunming Ye. 2012. Har: Hub, authority and relevance scores in multi-relational data for query search. In *SDM*, pages 141–152.

Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on multi-source multilingual information extraction and summarization*, pages 17–24. Association for Computational Linguistics.

Z. Liu, P. Li, Y. Zheng, and M. Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of EMNLP*, pages 257–266.

Z. Liu, W. Huang, Y. Zheng, and M. Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of EMNLP*, pages 366–376.

R. Mihalcea and P. Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, pages 404–411.

M.K.P. Ng, X. Li, and Y. Ye. 2011. Multirank: co-ranking for objects and relations in multi-relational data. In *Proceedings of the 17th ACM SIGKDD*, pages 1217–1225.

Paul Over and James Yen. 2004. Introduction to duc-2001: an intrinsic evaluation of generic news text summarization systems. In *Proceedings of DUC 2004 Document Understanding Workshop, Boston*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: bringing order to the web.

Peter D Turney. 1999. Learning to extract keyphrases from text. national research council. *Institute for Information Technology, Technical Report ERB-1057*.

X. Wan and J. Xiao. 2008a. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of AAAI*, pages 855–860.

Xiaojun Wan and Jianguo Xiao. 2008b. Collabrank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 969–976. Association for Computational Linguistics.

X. Wan, J. Yang, and J. Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *ACL*, page 552.