Automatic Keyphrase Extraction by Bridging Vocabulary Gap *

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

Abstract

Keyphrase extraction aims to select a set of terms from a document as a short summary of the document. Most methods extract keyphrases according to their statistical properties in the given document. Appropriate keyphrases, however, are not always statistically significant or even do not appear in the given document. This makes a large *vocabulary gap* between a document and its keyphrases. In this paper, we consider that a document and its keyphrases both describe the same object but are written in two different languages. By regarding keyphrase extraction as a problem of translating from the language of documents to the language of keyphrases, we use word alignment models in statistical machine translation to learn translation probabilities between the words in documents and the words in keyphrases. According to the translation model, we suggest keyphrases given a new document. The suggested keyphrases are not necessarily statistically frequent in the document, which indicates that our method is more flexible and reliable. Experiments on news articles demonstrate that our method outperforms existing unsupervised methods on precision, recall and F-measure.

1 Introduction

Information on the Web is emerging with the development of Internet. It is becoming more and more important to effectively search and manage information. Keyphrases, as a brief summary of a document, provide a solution to help organize and retrieve documents, which have been widely used in digital libraries and information retrieval (Turney, 2000; Nguyen and Kan, 2007). Due to the explosion of information, it is ineffective for professional human indexers to manually annotate documents with keyphrases. How to automatically extract keyphrases from documents becomes an important research problem, which is usually referred to as *keyphrase extraction*.

Most methods for keyphrase extraction try to extract keyphrases according to their statistical properties. These methods are susceptible to low performance because many appropriate keyphrases may not be statistically frequent or even not appear in the document, especially for short documents. We name the phenomenon as the *vocabulary gap* between documents and keyphrases. For example, a research paper talking about "machine transliteration" may less or even not mention the phrase "machine translation". However, since "machine transliteration" is a sub-field of "machine translation", the phrase "machine translation" is also reasonable to be suggested as a keyphrase to indicate the topics of this paper. Let us take another example: in a news article talking about "iPad" and "iPhone", the word "Apple" may rarely ever come up. However, it is known that both "iPad" and "iPhone" are the products of "Apple", and the word "Apple" may thus be a proper keyphrase of this article.

We can see that, the essential challenge of keyphrase extraction is the vocabulary gap between documents and keyphrases. Therefore, the task of keyphrase extraction is how to capture the semantic relations between the words in documents and in keyphrases so as to bridge the vocabulary gap. In this paper, we provide a new perspective to

[*] Zhiyuan Liu and Xinxiong Chen have equal contribution to this work.

Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pages 135–144,
Portland, Oregon, USA, 23–24 June 2011. ©2011 Association for Computational Linguistics

documents and their keyphrases: each document and its keyphrases are descriptions to the same object, but the document is written using one language, while keyphrases are written using another language. Therefore, keyphrase extraction can be regarded as a translation problem from the language of documents into the language of keyphrases.

Based on the idea of translation, we use word alignment models (WAM) (Brown et al., 1993) in statistical machine translation (SMT) (Koehn, 2010) and propose a unified framework for keyphrase extraction: (1) From a collection of translation pairs of two languages, WAM learns translation probabilities between the words in the two languages. (2) According to the translation model, we are able to bridge the vocabulary gap and succeed in suggesting appropriate keyphrases, which may not necessarily frequent in their corresponding documents.

As a promising approach to solve the problem of vocabulary gap, SMT has been widely exploited in many applications such as information retrieval (Berger and Lafferty, 1999; Karimzadehgan and Zhai, 2010), image and video annotation (Duygulu et al., 2002), question answering (Berger et al., 2000; Echihabi and Marcu, 2003; Murdock and Croft, 2004; Soricut and Brill, 2006; Xue et al., 2008), query expansion and rewriting (Riezler et al., 2007; Riezler et al., 2008; Riezler and Liu, 2010), summarization (Banko et al., 2000), collocation extraction (Liu et al., 2009b; Liu et al., 2010b) and paraphrasing (Quirk et al., 2004; Zhao et al., 2010). Although SMT is a widely adopted solution to vocabulary gap, for various applications using SMT, the crucial and non-trivial problem is to find appropriate and enough translation pairs for SMT.

The most straightforward translation pairs for keyphrase extraction is document-keyphrase pairs. In practice, however, it is time-consuming to annotate a large collection of documents with keyphrases for sufficient WAM training. In order to solve the problem, we use titles and summaries to build translation pairs with documents. Titles and summaries are usually accompanying with the corresponding documents. In some special cases, titles or summaries may be unavailable. We are also able to extract one or more important sentences from the corresponding documents to construct sufficient translation pairs.

2 State of the Art

Some researchers (Frank et al., 1999; Witten et al., 1999; Turney, 2000) regarded keyphrase extraction as a binary classification problem (is-keyphrase or non-keyphrase) and learned models for classification using training data. These supervised methods need manually annotated training set, which is time-consuming. In this paper, we focus on unsupervised methods for keyphrase extraction.

The most simple unsupervised method for keyphrase extraction is using TFIDF (Salton and Buckley, 1988) to rank the candidate keyphrases and select the top-ranked ones as keyphrases. TFIDF ranks candidate keyphrases only according to their statistical frequencies, which thus fails to suggest keyphrases with low frequencies.

Starting with TextRank (Mihalcea and Tarau, 2004), graph-based ranking methods are becoming the state-of-the-art methods for keyphrase extraction (Liu et al., 2009a; Liu et al., 2010a). Given a document, TextRank first builds a word graph, in which the links between words indicate their semantic relatedness, which are estimated by the word co-occurrences in the document. By executing PageRank (Page et al., 1998) on the graph, we obtain the PageRank score for each word to rank candidate keyphrases.

In TextRank, a low-frequency word will benefit from its high-frequency neighbor words and thus be ranked higher as compared to using TFIDF. This alleviates the problem of vocabulary gap to some extent. TextRank, however, still tends to extract high-frequency words as keyphrases because these words have more opportunities to get linked with other words and obtain higher PageRank scores. Moreover, TextRank usually constructs a word graph simply according to word co-occurrences as an approximation of the semantic relations between words. This will introduce much noise because of connecting semantically unrelated words and highly influence extraction performance.

Some methods have been proposed to improve TextRank, of which ExpandRank (Wan and Xiao, 2008b; Wan and Xiao, 2008a) uses a small number, namely *k*, of neighbor documents to provide more information of word relatedness for the construction of word graphs. Compared to TextRank, ExpandRank performs better when facing the vocabulary gap by borrowing the information on *document level*. However, the finding of neighbor documents are usually arbitrary. This process may introduce much noise and result in *topic drift* when the document and its so-called neighbor documents are not exactly talking about the same topics.

Another potential approach to alleviate vocabulary gap is latent topic models (Landauer et al., 1998; Hofmann, 1999; Blei et al., 2003), of which latent Dirichlet allocation (LDA) (Blei et al., 2003) is most popular. Latent topic models learn topics from a collection of documents. Using a topic model, we can represent both documents and words as the distributions over latent topics. The semantic relatedness between a word and a document can be computed using the cosine similarities of their topic distributions. The similarity scores can be used as the ranking criterion for keyphrase extraction (Heinrich, 2005; Blei and Lafferty, 2009). On one hand, latent topic models use topics instead of statistical properties of words for ranking, which abates the vocabulary gap problem on *topic level*. On the other hand, the learned topics are usually very coarse, and topic models tend to suggest general words for a given document. Therefore, the method usually fails to capture the specific topics of the document.

In contract to the above-mentioned methods, our method addresses vocabulary gap on *word level*, which prevents from topic drift and works out better performance. In experiments, we will show our method can better solve the problem of vocabulary gap by comparing with TFIDF, TextRank, ExpandRank and LDA.

3 Keyphrase Extraction by Bridging Vocabulary Gap Using WAM

First, we give a formal definition of keyphrase extraction: given a collection of documents D, for each document $d \in D$, keyphrase extraction aims to rank candidate keyphrases according to their likelihood given the document d, i.e., $\Pr(p|d)$ for all $p \in P$, where P is the candidate keyphrase set. Then we select top-M_d ones as keyphrases, where M_d can be fixed or automatically determined by the system. The document d can be regarded as a sequence of words $\mathbf{w}_d = \{w_i\}_1^{N_d}$, where N_d is the length of d.

In Fig. 1, we demonstrate the framework of keyphrase extraction using WAM. We divide the algorithm into three steps: preparing translation pairs, training translation models and extracting keyphrases for a given document. We will introduce the three steps in details from Section 3.1 to Section 3.3.

Input: A large collection of documents D for keyphrase extraction.

Step 1: Prepare Translation Pairs. For each $d \in D$, we may prepare two types of translation pairs:

- **Title-based Pairs**. Use the title t_d of each document d and prepare translation pairs, denote as $\langle D, T \rangle$.
- **Summary-based Pairs**. Use the summary s_d of each document d and prepare translation pairs, denote as $\langle D, S \rangle$.

Step 2: Train Translation Model. Given translation pairs, e.g., $\langle D, T \rangle$, train word-word translation model $\Pr_{\langle D,T \rangle}(t|w)$ using WAM, where w is the word in document language and t is the word in title language.

Step 3: Keyphrase Extraction. For a document d, extract keyphrases according to a trained translation model, e.g., $\Pr_{\langle D,T \rangle}(t|w)$.

1. Measure the importance score $\Pr(w|d)$ of each word w in document d.
2. Compute the ranking score of candidate keyphrase p by

$$\Pr(p|d) = \textstyle\sum_{t \in p} \sum_{w \in d} \Pr_{\langle D,T \rangle}(t|w) \Pr(w|d) \qquad (1)$$

3. Select top-M_d ranked candidate keyphrases according to $\Pr(p|d)$ as the keyphrases of document d.

Figure 1: WAM for keyphrase extraction.

3.1 Preparing Translation Pairs

Training dataset for WAM consists of a number of translation pairs written in two languages. In keyphrase extraction task, we have to construct sufficient translation pairs to capture the semantic relations between documents and keyphrases. Here we propose to construct two types of translation pairs: title-based pairs and summary-based pairs.

3.1.1 Title-based Pairs

Title is usually a short summary of the given document. In most cases, documents such as research papers and news articles have corresponding titles. Therefore, we can use title to construct translation pairs for a document.

WAM assumes each translation pair should be of comparable length. However, a document is usually much longer than title. It will hurt the performance if we fill the length-unbalanced pairs for WAM training. We propose two methods to address the problem: sampling method and split method.

In sampling method, we perform word sampling for each document to make it comparable to the length of its title. Suppose the lengths of a document and its title are N_d and N_t, respectively. For document d, we first build a bag of words $\mathbf{b}_d = \{(w_i, e_i)\}_{i=1}^{W_d}$, where W_d is the number of *unique* words in d, and e_i is the weights of word w_i in d.

In this paper, we use TFIDF scores as the weights of words. Using $\mathbf{b}_d$, we sample words for N_t times with replacement according to the weights of words, and finally form a new bag with N_t words to represent document d. In the sampling result, we keep the most important words in document d. We can thus construct a document-title pair with balanced length.

In split method, we split each document into sentences which are of comparable length to its title. For each sentence, we compute its semantic similarity with the title. There are various methods to measure semantic similarities. In this paper, we use vector space model to represent sentences and titles, and use cosine scores to compute similarities. If the similarity is smaller than a threshold δ, we will discard the sentence; otherwise, we will regard the sentence and title as a translation pair.

Sampling method and split method have their own characteristics. Compared to split method, sampling method loses the order information of words in documents. While split method generates much more translation pairs, which leads to longer training time of WAM. In experiment section, we will investigate the performance of the two methods.

3.1.2 Summary-based Pairs

For most research articles, authors usually provide abstracts to summarize the articles. Many news articles also have short summaries. Suppose each document itself has a short summary, we can use the summary and document to construct translation pairs using either sampling method or split method. Because each summary usually consists of multiple sentences, split method for constructing summary-based pairs has to split both the document and summary into sentences, and the sentence pairs with similarity scores above the threshold are filled in training dataset for WAM.

3.2 Training Translation Models

Without loss of generality, we take title-based pairs as the example to introduce the training process of translation models, and suppose documents are written in one language and titles are written in another language. In this paper, we use IBM Model-1 (Brown et al., 1993) for WAM training. IBM Model-1 is a widely used word alignment algorithm which does not require linguistic knowledge for two languages [1].

In IBM Model-1, for each translation pair $\langle \mathbf{w}_d, \mathbf{w}_t \rangle$, the relationship of the document language $\mathbf{w}_d = \{w_i\}_{i=0}^{L_d}$ and the title language $\mathbf{w}_t = \{t_i\}_{i=0}^{L_t}$ is connected via a hidden variable $\mathbf{a} = \{a_i\}_{i=1}^{L_d}$ describing an alignment mapping from words of documents to words of titles,

$$\Pr(\mathbf{w}_d|\mathbf{w}_t) = \sum_{\mathbf{a}} \Pr(\mathbf{w}_d, \mathbf{a}|\mathbf{w}_t) \qquad (2)$$

For example, $a_j = i$ indicates word w_j in $\mathbf{w}_d$ at position j is aligned to word t_i in $\mathbf{w}_t$ at position i. The alignment $\mathbf{a}$ also contains empty-word alignments $a_j = 0$ which align words of documents to an empty word. IBM Model-1 can be trained using Expectation-Maximization (EM) algorithm (Dempster et al., 1977) in an unsupervised fashion. Using IBM Model-1, we can obtain the translation probabilities of two language-vocabularies, i.e., $\Pr(t|w)$ and $\Pr(w|t)$, where w is a word in document vocabulary and t is a word in title vocabulary.

IBM Model-1 will produce one-to-many alignments from one language to another language, and the trained model is thus asymmetric. Hence, we can

[1] We have also employed more sophisticated WAM algorithms such as IBM Model-3 for keyphrase extraction. However, these methods did not achieve better performance than the simple IBM Model-1. Therefore, in this paper we only demonstrate the experimental results using IBM Model-1.

train two different translation models by assigning translation pairs in two directions, i.e., (document → title) and (title → document). We denote the former model as $\Pr_{\mathrm{d2t}}$ and the latter as $\Pr_{\mathrm{t2d}}$. We define $\Pr_{\langle D,T\rangle}(t|w)$ in Eq.(1) as the harmonic mean of the two models:

$$\Pr_{\langle D,T\rangle}(t|w) \propto \left(\frac{\lambda}{\Pr_{\mathrm{d2t}}(t|w)} + \frac{(1-\lambda)}{\Pr_{\mathrm{t2d}}(t|w)}\right)^{-1} \quad (3)$$

where λ is the harmonic factor to combine the two models. When $\lambda = 1.0$ or $\lambda = 0.0$, it simply uses model $\Pr_{\mathrm{d2t}}$ or $\Pr_{\mathrm{t2d}}$, correspondingly. Using the translation probabilities $\Pr(t|w)$ we can bridge the vocabulary gap between documents and keyphrases.

3.3 Keyphrase Extraction

Given a document d, we rank candidate keyphrases by computing their likelihood $\Pr(p|d)$. Each candidate keyphrase p may be composed of multiple words. As shown in (Hulth, 2003), most keyphrases are noun phrases. Following (Mihalcea and Tarau, 2004; Wan and Xiao, 2008b), we simply select noun phrases from the given document as candidate keyphrases with the help of POS tags. For each word t, we compute its likelihood given d, $\Pr(t|d) = \sum_{w\in d}\Pr(t|w)\Pr(w|d)$, where $\Pr(w|d)$ is the weight of the word w in d, which is measured using normalized TFIDF scores. $\Pr(t|w)$ is the translation probabilities obtained from WAM training.

Using the scores of all words in candidate keyphrases, we compute the ranking score of each candidate keyphrase by summing up the scores of each word in the candidate keyphrase, i.e., $\Pr(p|d) = \sum_{t\in p}\Pr(t|d)$. In all, the ranking scores of candidate keyphrases is formalized in Eq. (1) of Fig. 1. According to the ranking scores, we can suggest top-M_d ranked candidates as the keyphrases, where M_d is the number of suggested keyphrases to the document d pre-specified by users or systems. We can also consider the number of words in the candidate keyphrase as a normalization factor to Eq. (1), which will be our future work.

4 Experiments

To perform experiments, we crawled a collection of $13,702$ Chinese news articles [2] from www.163.com, one of the most popular news websites in China. The news articles are composed of various topics including science, technology, politics, sports, arts, society and military. All news articles are manually annotated with keyphrases by website editors, and all these keyphrases come from the corresponding documents. Each news article is also provided with a title and a short summary.

In this dataset, there are $72,900$ unique words in documents, and $12,405$ unique words in keyphrases. The average lengths of documents, titles and summaries are 971.7 words, 11.6 words, and 45.8 words, respectively. The average number of keyphrases for each document is 2.4. In experiments, we use the annotated titles and summaries to construct translation pairs.

In experiments, we select GIZA++ [3] (Och and Ney, 2003) to train IBM Model-1 using translation pairs. GIZA++, widely used in various applications of statistical machine translation, implements IBM Models 1-5 and an HMM word alignment model.

To evaluate methods, we use the annotated keyphrases by www.163.com as the standard keyphrases. If one suggested keyphrase exactly matches one of the standard keyphrases, it is a correct keyphrase. We use precision $p = c_{correct}/c_{method}$, recall $r = c_{correct}/c_{standard}$ and F-measure $f = 2pr/(p+r)$ for evaluation, where $c_{correct}$ is the number of keyphrases correctly suggested by the given method, c_{method} is the number of suggested keyphrases, and $c_{standard}$ is the number of standard keyphrases. The following experiment results are obtained by 5-fold cross validation.

4.1 Evaluation on Keyphrase Extraction

4.1.1 Performance Comparison and Analysis

We use four representative unsupervised methods as baselines for comparison: TFIDF, TextRank (Mihalcea and Tarau, 2004), ExpandRank (Wan and Xiao, 2008b) and LDA (Blei et al., 2003). We denote our method as WAM for short.

In Fig. 2, we demonstrate the precision-recall curves of various methods for keyphrase extraction including TFIDF, TextRank, ExpandRank, LDA and WAM with title-based pairs prepared using

[2] The dataset can be obtained from http://nlp.csai.tsinghua.edu.cn/~lzy/datasets/ke_wam.html.

[3] The website for GIZA++ package is http://code.google.com/p/giza-pp/.

sampling method (Title-Sa) and split method (Title-Sp), and WAM with summary-based pairs prepared using sampling method (Summ-Sa) and split method (Summ-Sp). For WAM, we set the harmonic factor $\lambda = 1.0$ and threshold $\delta = 0.1$, which is the optimal setting as shown in the later analysis on parameter influence. For TextRank, LDA and ExpandRank, we report their best results after parameter tuning, e.g., the number of topics for LDA is set to 400, and the number of neighbor documents for ExpandRank is set to 5 .

The points on a precision-recall curve represent different numbers of suggested keyphrases from $M_d = 1$ (bottom right) to $M_d = 10$ (upper left), respectively. The closer the curve is to the upper right, the better the overall performance of the method is. In Table 1, we further demonstrate the precision, recall and F-measure scores of various methods when $M_d = 2$ [4]. In Table 1, we also show the statistical variances after ±. From Fig. 2 and Table 1, we have the following observations:

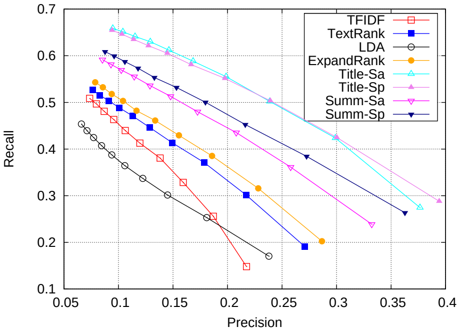

Figure 2: The precision-recall curves of various methods for keyphrase extraction.

First, our method outperforms all baselines. It indicates that the translation perspective is valid for keyphrase extraction. When facing vocabulary gap, TFIDF and TextRank have no solutions, ExpandRank adopts the external information on document level which may introduce noise, and LDA adopts the external information on topic level which may be too coarse. In contrast to these baselines, WAM aims to bridge the vocabulary gap on *word level*, which avoids topic drift effectively. Therefore, our method can better solve the problem of vocabulary gap in keyphrase extraction.

Method	Precision	Recall	F-measure
TFIDF	0.187	0.256	0.208±0.005
TextRank	0.217	0.301	0.243±0.008
LDA	0.181	0.253	0.203±0.002
ExpandRank	0.228	0.316	0.255±0.007
Title-Sa	0.299	0.424	0.337±0.008
Title-Sp	**0.300**	**0.425**	**0.339±0.010**
Summ-Sa	0.258	0.361	0.289±0.009
Summ-Sp	0.273	0.384	0.307±0.008

Table 1: Precision, recall and F-measure of various methods for keyphrase extraction when $M_d = 2$.

Second, WAM with title-based pairs performs better than summary-based pairs consistently, no matter prepared using sampling method or split method. This indicates the titles are closer to the keyphrase language as compared to summaries. This is also consistent with the intuition that titles are more important than summaries. Meanwhile, we can save training efforts using title-based pairs.

Last but not least, split method achieves better or comparable performance as compared to sampling method on both title-based pairs and summary-based pairs. The reasons are: (1) the split method generates more translation pairs for adequate training than sampling method; and (2) split method also keeps the context of words, which helps to obtain better word alignment, unlike bag-of-words in sampling method.

4.1.2 Influence of Parameters

We also investigate the influence of parameters to WAM with title-based pairs prepared using split method, which achieves the best performance as shown in Fig. 2. The parameters include: harmonic factor λ (described in Eq. 3) and threshold factor δ. Harmonic factor λ controls the weights of the translation models trained in two directions, i.e., $\Pr_{\mathrm{d2t}}(t|w)$ and $\Pr_{\mathrm{t2d}}(t|w)$ as shown in Eq. (3). As described in Section 3.1.1, using threshold factor δ we filter out the pairs with similarities lower than δ.

In Fig. 3, we show the precision-recall curves of WAM for keyphrase extraction when harmonic factor λ ranges from 0.0 to 1.0 stepped by 0.2. From the figure, we observe that the translation model $\Pr_{\mathrm{d2t}}(t|w)$ (i.e., when $\lambda = 1.0$) performs better than

[4] We select $M_d = 2$ because WAM gains the best F-measure score when $M_d = 2$, which is close to the average number of annotated keyphrases for each document 2.4.

$\mathrm{Pr}_{\mathrm{t2d}}(t|w)$ (i.e., when $\lambda = 0.0$). This indicates that it is sufficient to simply train a translation model in one direction (i.e., $\mathrm{Pr}_{\mathrm{d2t}}(t|w)$) for keyphrase extraction.

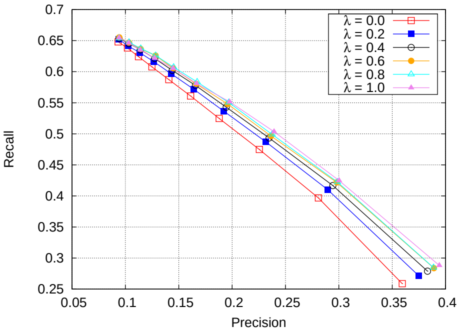

Figure 3: Precision-recall curves of WAM when harmonic factor λ ranges from 0.0 to 1.0.

In Fig. 4, we show the precision-recall curves of WAM for keyphrase extraction when threshold factor δ ranges from 0.01 to 0.90. In title-based pairs using split method, the total number of pairs without filtering any pairs (i.e., $\delta = 0$) is $347,188$. When $\delta = 0.01$, 0.10 and 0.90, the numbers of retained translation pairs are $165,023$, $148,605$ and $41,203$, respectively. From Fig. 4, we find that more translation pairs result in better performance. However, more translation pairs also indicate more training time of WAM. Fortunately, we can see that the performance does not drop much when discarding more translation pairs with low similarities. Even when $\delta = 0.9$, our method can still achieve performance with precision $p = 0.277$, recall $r = 0.391$ and F-measure $f = 0.312$ when $M_d = 2$. Meanwhile, we reduce the training efforts by about 50% as compared to $\delta = 0.01$.

In all, based on the above analysis on two parameters, we demonstrate the effectiveness and robustness of our method for keyphrase extraction.

4.1.3 When Titles/Summaries Are Unavailable

Suppose in some special cases, the titles or summaries are unavailable, how can we construct translation pairs? Inspired by extraction-based document summarization (Goldstein et al., 2000; Mihalcea and Tarau, 2004), we can extract one or more important sentences from the given document to construct translation pairs. Unsupervised sentence extraction for document summarization is a well-studied task in natural language processing. As shown in Table 2, we only perform two simple sentence extraction methods to demonstrate the effectiveness: (1) Select the first sentence of a document (denoted as "First"); and (2) Compute the cosine similarities between each sentence and the whole document represented as two bags-of-words (denoted as "Importance").

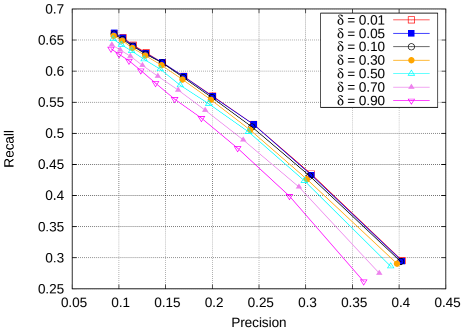

Figure 4: Precision-recall curves of WAM when threshold δ ranges from 0.01 to 0.90.

It is interesting to find that the method of using the first sentence performs similar to using titles. This profits from the characteristic of news articles which tend to give a good summary for the whole article using the first sentence. Although the second method drops much on performance as compared to using titles, it still outperforms than other existing methods. Moreover, the second method will improve much if we use more effective measures to identify the most important sentence.

Method	Precision	Recall	F-measure
First	0.290	0.410	0.327±0.013
Importance	0.260	0.367	0.293±0.010

Table 2: Precision, recall and F-measure of keyphrase extraction when $M_d = 2$ by extracting one sentence to construct translation pairs.

4.2 Beyond Extraction: Keyphrase Generation

In Section 4.1, we evaluate our method on keyphrase extraction by suggesting keyphrases from documents. In fact, our method is also able to suggest keyphrases that have not appeared in the content of given document. The ability is important especially when the length of each document is short, which itself may not contain appropriate keyphrases. We name the new task *keyphrase generation*. To evaluate these methods on keyphrase generation, we perform keyphrase generation for the titles of documents, which are usually much shorter than their corresponding documents. The experiment setting is as follows: the training phase is the same to the previous experiment, but in the test phase we suggest keyphrases only using the titles. LDA and ExpandRank, similar to our method, are also able to select candidate keyphrases beyond the titles. We still use the annotated keyphrases of the corresponding documents as standard answers. In this case, about 59% standard keyphrases do not appear in titles.

In Table 3 we show the evaluation results of various methods for keyphrase generation when $M_d = 2$. For WAM, we only show the results using title-based pairs prepared with split method. From the table, we have three observations: (1) WAM outperforms other methods on keyphrase generation. Moreover, there are about 10% correctly suggested keyphrases by WAM do not appear in titles, which indicates the effectiveness of WAM for keyphrase generation. (2) The performance of TFIDF and TextRank is much lower as compared to Table 1, because the titles are so short that they do not provide enough candidate keyphrases and even the statistical information to rank candidate keyphrases. (3) LDA, ExpandRank and WAM roughly keep comparable performance as in Table 1 (The performance of ExpandRank drops a bit). This indicates the three methods are able to perform keyphrase generation, and verifies again the effectiveness of our method.

Method	Precision	Recall	F-measure
TFIDF	0.105	0.141	0.115±0.004
TextRank	0.107	0.144	0.118±0.005
LDA	0.180	0.256	0.204±0.008
ExpandRank	0.194	0.268	0.216±0.012
WAM	**0.296**	**0.420**	**0.334±0.009**

Table 3: Precision, recall and F-measure of various methods for keyphrase generation when $M_d = 2$.

To demonstrate the features of our method for keyphrase generation, in Table 4 we list top-5 keyphrases suggested by LDA, ExpandRank and WAM for a news article entitled *Israeli Military Claims Iran Can Produce Nuclear Bombs and Considering Military Action against Iran* (We translate the original Chinese title and keyphrases into English for comprehension.). We have the following observations: (1) LDA suggests general words like "negotiation" and "sanction" as keyphrases because the coarse-granularity of topics. (2) ExpandRank suggests some irrelevant words like "Lebanon" as keyphrases, which are introduced by neighbor documents talking about other affairs related to Israel. (3) Our method can generate appropriate keyphrases with less topic-drift. Moreover, our method can find good keyphrases like "nuclear weapon" which even do not appear in the title.

LDA: Iran, U.S.A., negotiation, Israel, sanction
ExpandRank: Iran, Israel, Lebanon, U.S.A., Israeli Military
WAM: Iran, military action, Israeli Military, Israel, nuclear weapon

Table 4: Top-5 keyphrases suggested by LDA, ExpandRank and WAM.

5 Conclusion and Future Work

In this paper, we provide a new perspective to keyphrase extraction: regarding a document and its keyphrases as descriptions to the same object written in two languages. We use IBM Model-1 to bridge the vocabulary gap between the two languages for keyphrase generation. We explore various methods to construct translation pairs. Experiments show that our method can capture the semantic relations between words in documents and keyphrases. Our method is also language-independent, which can be performed on documents in any languages.

We will explore the following two future work: (1) Explore our method on other types of articles and on other languages. (2) Explore more complicated methods to extract important sentences for constructing translation pairs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 60873174. The authors would like to thank Peng Li and Xiance Si for their suggestions.

References

M. Banko, V.O. Mittal, and M.J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of ACL*, pages 318–325.

A. Berger and J. Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of SIGIR*, pages 222–229.

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of SIGIR*, pages 192–199.

D.M. Blei and J.D. Lafferty, 2009. *Text mining: Classification, Clustering, and Applications*, chapter Topic models. Chapman & Hall.

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

P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

A.P. Dempster, N.M. Laird, D.B. Rubin, et al. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

P. Duygulu, Kobus Barnard, J. F. G. de Freitas, and David A. Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of ECCV*, pages 97–112.

A. Echihabi and D. Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of ACL*, pages 16–23.

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

J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48.

G. Heinrich. 2005. Parameter estimation for text analysis. *Web: http://www. arbylon. net/publications/text-est.*

T. Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, pages 50–57.

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

M. Karimzadehgan and C.X. Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of SIGIR*, pages 323–330.

P. Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

T.K. Landauer, P.W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

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

Z. Liu, H. Wang, H. Wu, and S. Li. 2009b. Collocation extraction using monolingual word alignment method. In *Proceedings of EMNLP*, pages 487–495.

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

Z. Liu, H. Wang, H. Wu, and S. Li. 2010b. Improving statistical machine translation with monolingual collocation. In *Proceedings of ACL*, pages 825–833.

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

V. Murdock and W.B. Croft. 2004. Simple translation models for sentence retrieval in factoid question answering. In *Proceedings of SIGIR*.

T. Nguyen and M.Y. Kan. 2007. Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries*, pages 317–326.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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

C. Quirk, C. Brockett, and W. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, volume 149.

S. Riezler and Y. Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics*, 36(3):569–582.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proccedings of ACL*, pages 464–471.

S. Riezler, Y. Liu, and A. Vasserman. 2008. Translating queries into snippets for improved query expansion. In *Proceedings of COLING*, pages 737–744.

G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing and management*, 24(5):513–523.

R. Soricut and E. Brill. 2006. Automatic question answering using the web: Beyond the factoid. *Information Retrieval*, 9(2):191–206.

P.D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

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

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

I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of DL*, pages 254–255.

X. Xue, J. Jeon, and W.B. Croft. 2008. Retrieval models for question and answer archives. In *Proceedings of SIGIR*, pages 475–482.

S. Zhao, H. Wang, and T. Liu. 2010. Paraphrasing with search engine query logs. In *Proceedings of COLING*, pages 1317–1325.