


# Cross-language document summarization via extraction and ranking of multiple summaries

Xiaojun Wan<sup>1,2</sup>  · Fuli Luo<sup>2</sup> · Xue Sun<sup>1</sup> ·  
Songfang Huang<sup>3</sup> · Jin-ge Yao<sup>1</sup>

Received: 27 September 2016 / Revised: 9 August 2017 / Accepted: 4 January 2018  
© Springer-Verlag London Ltd., part of Springer Nature 2018

**Abstract** The task of cross-language document summarization aims to produce a summary in a target language (e.g., Chinese) for a given document set in a different source language (e.g., English). Previous studies focus on ranking and selection of translated sentences in the target language. In this paper, we propose a new framework for addressing the task by extraction and ranking of multiple summaries in the target language. First, we extract multiple candidate summaries by proposing several schemes for improving the upper-bound quality of the summaries. Then, we propose a new ensemble ranking method for ranking the candidate summaries by making use of bilingual features. Extensive experiments have been conducted on a benchmark dataset and the results verify the effectiveness of our proposed framework, which outperforms a variety of baselines, including supervised baselines.

**Keywords** Document summarization · Natural language generation · Natural language processing · Text mining

---

✉ Xiaojun Wan  
wanxiaojun@pku.edu.cn

Fuli Luo  
1213952436@qq.com

Xue Sun  
1300012839@pku.edu.cn

Songfang Huang  
huangsf@cn.ibm.com

Jin-ge Yao  
yaojing@pku.edu.cn

<sup>1</sup> Institute of Computer Science and Technology, Peking University, Beijing, China

<sup>2</sup> Key Laboratory of Computational Linguistics (Peking University), MOE, Beijing, China

<sup>3</sup> IBM China Research Laboratory, Beijing, China

# 1 Introduction

Cross-language document summarization is the task of producing a summary in a target language from documents in a different source language. The task is a particularly useful extension of traditional monolingual document summarization in the multilingual environment. In this study, we take English-to-Chinese cross-language summarization as a use case, as in previous studies [22, 27]. Specifically, given an English document set about a topic, we can utilize the cross-language document summarization system to produce a short Chinese summary. The produced Chinese summary is useful for many Chinese readers to quickly understand the English topic, especially for those readers who do not know or are not very familiar with English.

There are only a few previous studies investigating this special document summarization task, and most of them rely on ranking and selection of translated sentences in the target language. In particular, Wan et al. [23] proposed to rank and select sentences by fusing two kinds of sentence-level scores: the informativeness and the quality. Wan [22] proposed a co-ranking framework to simultaneously rank both sentences in the source language and sentences in the target language and then select the target-side sentences according to the ranking scores. Yao et al. [27] further proposed a phrase-based model for scoring the target-side sentences and then selecting the summary sentences in a greedy way. During selection, a sentence may be compressed by dropping some phrases.

However, all the above methods involve with target-side sentence ranking and selection, and thus, the quality of a cross-language summary largely depends on the sentence ranking results. However, a summary's overall quality is reflected by many different factors, including a large number of word-level, sentence-level, summary-level factors. Previous methods cannot consider the multiple factors to assess the summary's quality and guide the summary extraction process, no matter they adopt greedy algorithms or global inference methods (e.g., integer linear programming—ILP) for the sentence selection.

Moreover, all the methods make use of the same cross-language summarization model (i.e., a specific cross-language summarization method with a specific parameter setting) for summarizing different document sets in the source language. For example, Wan [22] made use of the co-ranking method with  $\lambda = 0.8$  and  $\alpha = 0.5$  for summarization of all the 30 document sets. However, as we show later, different document sets usually have different characteristics. A single summarization model may not produce high-quality summary for every document set, even when the model may lead to good average summarization performance across all document sets.

In order to address the above problems, we propose a new framework for addressing the cross-language document summarization task by extraction and ranking of multiple summaries in the target language. First, we extract multiple candidate summaries with different summarization models for each document set and propose several strategies to make the candidate summaries for each document set contain some high-quality summaries. Then, we explore learning to rank techniques and further propose a top- $K$  ensemble ranking method to rank the candidate summaries for each document set by making full use of multiple features of different levels. The top-ranked summary is used as the final cross-language summary for each document set. Extensive experiments have been conducted on a benchmark dataset and the results verify the effectiveness of our proposed cross-language summarization framework, which outperforms a variety of baselines, including unsupervised and supervised baselines. The proposed ensemble ranking method has proved to be more effective than traditional learning to rank methods.

The contributions of this paper are summarized as follows:

1. We propose a new framework for addressing the challenging cross-language document summarization task by extraction and ranking of multiple summaries in the target language.
2. We propose several strategies to extract candidate summaries and improve the upper-bound quality of the candidate summary set.
3. We propose a top- $K$  ensemble ranking method for candidate summary ranking and investigate a number of features to characterize the quality of a candidate summary.
4. Evaluation results on a benchmark dataset validate the effectiveness of our proposed framework and the ensemble ranking method, and the usefulness of the multiple groups of features.

In the rest of this paper, we first introduce the problem and data in Sect. 2 and then describe our proposed framework in Sect. 3. The details of the two steps in our framework (i.e., candidate summary extraction and candidate summary ranking) are introduced in Sects. 4 and 5, respectively. Empirical evaluation results are presented and discussed in Sect. 6. In Sect. 7, we introduce related work in brief. Lastly, we conclude this paper in Sect. 8.

## 2 Problem and data

In this study, we focus on English-to-Chinese cross-language document summarization. The input is an English document set relevant to a topic, and the expected output is a Chinese summary with a predefined length limit. In order to produce a Chinese summary from English documents, we need an English-to-Chinese machine translation step. However, the machine translation quality is far from satisfactory, with the translation results containing many errors and noises. This makes the cross-language summarization task quite challenging because the errors and noises brought by machine translation have large negative impact on the evaluation of a sentence or a summary in the target language. As shown in previous studies [22, 27], it is not a good choice to simply use methods of either translating English summaries produced by an existing monolingual summarization method or directly summarizing the translated Chinese documents. Therefore, we need new methods to address this challenging task.

For a fair comparison, we use the benchmark dataset provided by Wan [22] as the test set. The dataset was built based on the test set provided by DUC 2001 by manually translating the reference English summaries into Chinese summaries and then treating the manually translated Chinese summaries as reference summaries for the cross-language summarization task. There are in total 30 English document sets in DUC 2001, and each document set has two or three generic reference summaries with a length limit. The test set has also been used for evaluation in Yao et al. [27]. Following the character budgeting scheme in Yao et al. [27], the length limit of the peer Chinese summaries is fixed to 300 Chinese characters for all document sets for a fair comparison.

Since our proposed framework is based on supervised learning, we need to annotate additional data. We employed three graduate students who are quite familiar with English to manually translate all the reference summaries for 89 document sets, consisting of 30 document sets in the training set provided by DUC 2001 and 59 document sets provided by DUC 2002. In the experiments, we used the 89 document sets for model training and parameter tuning.

Note that the cross-language summarization task investigated in this study is a totally different summarization task from the Multiling Pilot task in TAC, which consists of a batch of monolingual summarization tasks in different languages.

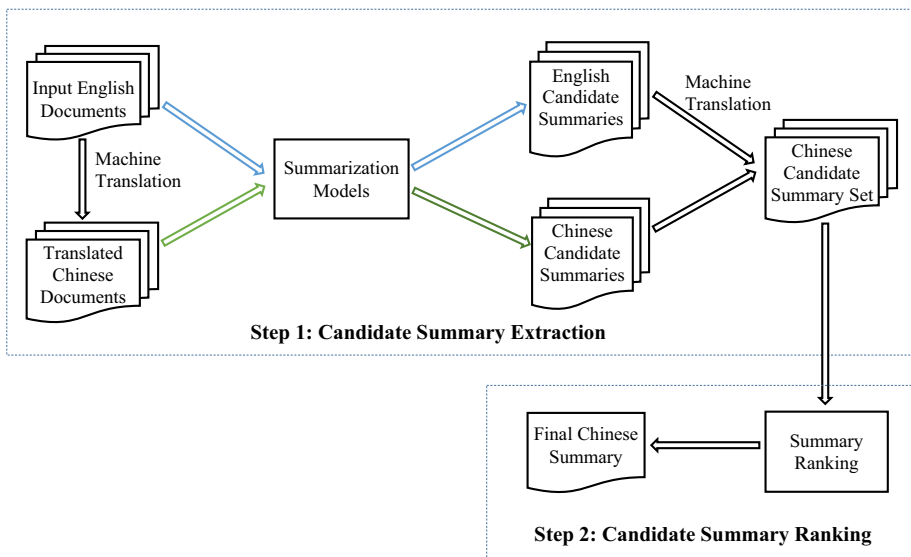
### 3 Our proposed framework

Our proposed framework consists of two major steps: candidate summary extraction and candidate summary ranking, as shown in Fig. 1.

The first step aims to produce a number of candidate Chinese summaries for each English document set. The candidate summaries are produced by applying a simple summarization method with different parameter settings on both the original English document set and the translated Chinese document set. Based on the translated Chinese document set, a set of Chinese candidate summaries can be extracted. Based on the original English document set, a set of English candidate summaries can be extracted, and then these summaries are automatically translated into Chinese by using machine translation. The final Chinese candidate summary set is a combination of the directly extracted Chinese candidate summaries and the translated Chinese candidate summaries. We believe the candidate set contains some high-quality summaries for the corresponding document set. We further consider multiple machine translation services and adopt sentence compression techniques to improve the upper-bound quality of the candidate summaries for each document set.

The second step aims to rank the candidate summaries produced for each document set by using learning to rank techniques. We expect the top-ranked summary to be high quality because we investigate and develop a number of features to characterize the quality of a candidate Chinese summary from different levels and perspectives. We further propose an ensemble ranking method to improve the ranking performance.

The details of the two steps are described in next sections, respectively.



**Fig. 1** Our proposed framework for cross-language document summarization

## 4 Candidate summary extraction

### 4.1 Method

In this step, we produce a number of candidate Chinese summaries for each English document set. There are several choices for achieving this goal. For example, we can make use of different summarization methods, a summarization method with different parameter settings, or both of them to produce different summaries for each document set. In this study, we simply use an off-the-shelf summarization method with different parameter settings for producing candidate summaries. More complex ways will certainly work in this step, which are not the focus of this paper.

Particularly, we adopt the summarization method proposed by Lin and Bilmes [14], which is based on budgeted maximization of a simple submodular function defined as follows:

$$f(S) = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j} - \lambda \sum_{i,j \in S: i \neq j} w_{i,j}$$

And the summarization problem is formalized as maximizing the submodular function under budget constraint:

$$\max_{S \subseteq V} \left\{ f(S) : \sum_{i \in S} c_i \leq B \right\}$$

where  $\lambda \geq 0$  is the weight of the redundancy penalty term.  $V$  is the set of all sentences,  $S$  is the extracted summary (a subset of  $V$ ),  $c_i$  is the cost of sentence  $i$ , i.e., the length of sentence  $i$ , and  $B$  is the budget constraint, i.e., the length limit of summary.  $w_{i,j}$  is the cosine similarity of sentences  $i$  and sentence  $j$ . In this study,  $B$  is set to 300 Chinese characters.  $c_i$  is the number of Chinese characters for a Chinese sentence, and it is the number of Chinese characters in the translated Chinese sentence for an English sentence.

The above problem can be solved by using a simple greedy algorithm with guaranteed near-optimality. The greedy algorithm (see Algorithm 1 in Lin and Bilmes [14]) sequentially finds sentence with the largest ratio of objective function grain to scaled cost, i.e.,  $(f(G \cup \{l\}) - f(G)) / c_l^r$ , where  $r > 0$  is the scaling factor for the cost of sentence  $l$ . If adding the sentence increases the objective function value while not violating the budget constraint, it is then selected and otherwise bypassed.

We can see that there are two parameters in the above summarization method:  $\lambda$  and  $r$ , where  $\lambda$  controls the weight of the redundancy penalty term, and  $r$  is the scaling factor in the greedy selection algorithm. In order to produce different candidate summaries for each document set, we simply vary  $\lambda$  from 1 to 6 with a step of 1, and vary  $r$  from 0.2 to 1.0 with a step of 0.2. We further set  $\lambda$  to 0, which means removing the penalty term, and vary  $r$  from 0.1 to 1.0 with a step of 0.1. In this way, we get a total of 40 different parameter settings, resulting in 40 more or less different candidate summaries for each document set. The submodular function-based summarization method with 40 different parameter settings can be applied in either English documents or translated Chinese documents. The extracted English summaries are then translated into Chinese with machine translation. Thus, we have 80 Chinese candidate summaries. Moreover, we propose the following three strategies to improve the candidate summary set.

**Bilingual submodular function** The above submodular function  $f(S)$  is denoted as  $f_{\text{cn}}(S)$  when it is applied in the Chinese documents and  $f_{\text{en}}(S)$  in the English documents. We can simply combine the two functions as follows:

$$f_{\text{bilingual}}(S) = \alpha f_{\text{cn}}(S) + (1 - \alpha) f_{\text{en}}(S)$$

where  $\alpha \in [0, 1]$  is the combination coefficient. We can use  $f_{\text{bilingual}}(S)$  in the greedy algorithm for summary extraction. By using this submodular function, the bilingual information of a sentence can be considered in the algorithm. Here, we fix  $\lambda$  to 6 and vary  $r$  from 0.2 to 1.0 with a step of 0.2,  $\alpha$  from 0.15 to 0.90 with a step of 0.15. In this way, we obtain additional 30 candidate Chinese summaries.

**Multiple machine translations** With a machine translation service, we can obtain 110 (40+40+30) Chinese candidate summaries by using all the above submodular functions with different parameter settings. Considering that different machine translation services usually have different translation results, we explore to use three state-of-the-art English-to-Chinese machine translation services: Google Translate,<sup>1</sup> Baidu Translate,<sup>2</sup> and Youdao Translate.<sup>3</sup> With each machine translation service, we can obtain 110 Chinese candidate summaries for each document set, and with all the three machine translation service, we can obtain a total of 330 Chinese candidate summaries.

**Multiple sentence compression** For multi-document summarization, similar sentences can be merged into one single sentence, which is a better choice for summary extraction. We further leverage takahe<sup>4</sup> [6] for multiple English sentence compression. takahe constructs a word graph by iteratively adding sentences to it, and the best compression is obtained by finding the shortest path in the word graph. We first use affinity propagation clustering [8] for sentence clustering and select redundant sentences by setting the ratio parameter to 0.7. takahe is then used to compress the redundant sentences into a single sentence, by which all the original sentences are replaced. After the compression, a small number of sentences in the English document set have been modified. We apply all the previous strategies for candidate summary extraction on the modified document set and obtain 330 Chinese candidate summaries. These 330 Chinese candidate summaries may highly overlap with the previous 330 Chinese candidate summaries produced before sentence compression. Therefore, we do not combine them together. Instead, we test them independently to verify the usefulness of sentence compression. Note that the total number (330) of candidate summaries is not so large, as compared to the total number of all possible hundreds of thousands of candidate summaries.

## 4.2 Analysis

It is expected that the candidate summary set contains some high-quality summaries for each document set, which makes the summary ranking process meaningful. So we first investigate the upper-bound performance of summary ranking on the test set. The upper-bound performance is the performance when we always select the best candidate summary from the candidate summary set for each document set. We used the ROUGE 1.5.5 toolkit [12] for performance evaluation. Before applying the ROUGE toolkit, we use the Stanford Chinese word segmenter with the CTB model for Chinese word segmentation of both reference summaries and candidate summaries. We then report F-scores of ROUGE-1 and ROUGE-2 based on the word segmentation results. Since a Chinese word often consists of two or more Chinese characters, we focus more on ROUGE-1.

<sup>1</sup> <http://translate.google.com/>.

<sup>2</sup> <http://fanyi.baidu.com/>.

<sup>3</sup> <http://fanyi.youdao.com/>.

<sup>4</sup> <https://github.com/boudinfl/takahe>.

**Table 1** ROUGE-1 comparison on candidate set

	# summaries	Upper bound	Average	Lower bound	Best model
Google	110	0.30895	0.26338	0.20949	0.28179
Baidu	110	0.30311	0.25674	0.20916	0.28279
Youdao	110	0.30823	0.26233	0.21343	0.28330
MultiTrans	330	0.31768	0.26081	0.19942	0.28330
Compression	330	0.31999	0.26324	0.19967	0.28370

**Table 2** ROUGE-2 comparison on candidate set

	# summaries	Upper bound	Average	Lower bound	Best model
Google	110	0.06018	0.03921	0.02034	0.04419
Baidu	110	0.06099	0.03961	0.02157	0.04621
Youdao	110	0.06254	0.04049	0.02288	0.04552
MultiTrans	330	0.06867	0.03977	0.01598	0.04552
Compression	330	0.07090	0.04073	0.01656	0.04456

Tables 1 and 2 present the comparison results on the candidate summary set with respect to ROUGE-1 and ROUGE-2, respectively. In the tables, in addition to the upper-bound performance of different candidate sets, we also report the average, the lower-bound performance of the candidate sets. The average performance means averaging the ROUGE scores of all the candidate summaries for each document set. The lower-bound performance corresponds to the performance when the worst summary is selected from the candidate set for each document set. We also present the performance of the best single model on the test set. For example, when using Google Translate for machine translation, the best single model is based on the submodular function  $f(S)$  with a specific parameter setting:  $\lambda = 6$  and  $r = 0.8$ ; while when using Baidu Translate for machine translation, the best single model is based on the submodular function  $f_{\text{bilingual}}(S)$  with  $\lambda = 6$ ,  $r = 0.4$  and  $\alpha = 0.75$ . Note that the best single model is found by exhaustive parameter searching on the test set. Therefore, the performance is only for reference. In real experiments, we should not find such “best” single model if we tune parameters on the training set.

As seen from the tables, there is a big gap between the upper-bound performance and the lower-bound performance, which means that the quality of the summaries in the candidate set is divergent. We can also see that the upper-bound performance is much better than that of the best model, which means that a single summarization model could not always produce high-quality summaries for all document sets. The results verify the necessity of the summary ranking process.

Furthermore, when comparing different strategies for producing candidate summaries, we find that the three machine translation services do not make much differences on the upper-bound performance. However, making use of all the three machine translation services may bring much improvement of the upper-bound performance. The strategy of multiple sentence compression can slightly improve the upper-bound performance. Overall, the use of multiple translations and sentence compression is beneficial for producing better candidate summaries. We adopt the 330 candidate summaries produced after utilizing all the proposed strategies as the candidate set for summary ranking.

## 5 Candidate summary ranking

### 5.1 Learning to rank

We explore learning to rank techniques for candidate summary ranking for each document set. We use various learning to ranking algorithms including Ranking SVM [11], RankNet [1], RankBoost [7], RandomForest [2], ListNet [3]. We find Ranking SVM is competitive and stable. So in this study, we utilize Ranking SVM for candidate summary ranking and use the popular SVM-rank tool in the experiments.

We need to build training data to train Ranking SVM. For each document set in the training set, the quality of each of the 330 candidate Chinese summaries is measured by comparing it with the reference Chinese summaries. We treat each document set as a “query” and use the ROUGE-1 score as the target value for each candidate summary and thus construct the training data for learning to rank. After training a ranking model, we apply the model to rank candidate summaries for each document set in the test set.

### 5.2 Top- $K$ ensemble ranking

In order to improve the summary ranking results, we further propose a top- $K$  ensemble ranking method. The basic idea of our proposed ensemble ranking method is to fish out the best of the best ones. We train  $P$  basic rankers (i.e., Ranking SVM) by sampling different subsets from the training set to rank the candidate summaries for each document set, and make use of the ranking positions as new features for training a new ensemble ranker, namely a re-ranker. While learning the ensemble ranker, we filter out those bad examples and keep only the best ones. If a candidate summary receives at least one top- $K$  position in the multiple ranking lists produced by the different basic rankers, the summary is treated as one of the best examples. Otherwise, the summary is treated as a bad example.  $K \in [1, 330]$  and  $P \in \mathbb{N}^+$  are parameters.

The algorithm is presented in Table 3. We use the multiple basic rankers to select best candidates from all candidate summaries, and then train a new ranker to fish out the best of the best ones. The multiple basic rankers also provide news features for the re-ranker.

**Table 3** Our top- $K$  ensemble ranking algorithm

Given: training set  $T$ , basic ranker—svmrnk,  $K$ ,  $P$ ;

Output: a re-ranking model;

Algorithm:

1. Randomly sample 60% document sets (with their candidate summaries) from  $T$  for  $P$  times;
2. Train  $P$  basic ranking models on the sampled data sets with the basic ranker, and apply the ranking models to the whole training set to obtain  $P$  ranking positions for each candidate summary for each document set;
3. Filter out those summaries which are bad examples, i.e., they never receive a top- $K$  position in the  $P$  ranking lists;
4. Construct new training data  $RT$  for re-ranking by using the  $P$  positions as  $P$  features for the remaining candidate summaries;
5. Train a re-ranking model on  $RT$  with the basic ranker;



After we get the re-ranking model, we apply the model on the test set to rank the candidate summaries for each document set. Note that we also need to apply the  $P$  basic rankers to get  $P$  ranking positions as features for each candidate summary.

### 5.3 Features

In order to better characterize the quality of a candidate summary, we develop a number of features from different levels and perspectives. They are divided into five groups: word-level features, sentence-level features, summary-level features, readability-related features and source-side features. Note that the first four groups of features are extracted on the Chinese-side, while the source-side features are the summary-level features extracted based on the English summary candidates and the English original document set.

*Word-level features* This group of features is extracted based on simple word-level information, such as the sum of TF, the sum of TFIDF, the word count, the ratio of stop words, punctuation marks, special POS tags (noun, verb and adjective), named entities, unique words (word types), words in the leading sentences.

*Sentence-level features* This group of features includes the mean/max/min length of sentences, the mean position weight, the mean cosine similarity with the leading sentences, and the mean PageRank values of sentences.

*Summary-level features* This group of features includes the cosine, Jaccard, Dice similarity between the summary and the document set, and between the summary and other candidate summaries. The summary-level PageRank scores are also computed and used.

*Readability-related features* We develop several simple features for measuring the readability of the Chinese summary, including conjunction score, the ratio of pronoun, the continuity score and the bigram probability.

*Source-side features* This group of features is extracted from the English side by using the same summary-level features. We do not use the word-level features and sentence-level features because they do not contribute to the results.

The detailed descriptions of features used in this study are presented in Tables 4, 5, 6 and 7, respectively.

## 6 Empirical evaluation

### 6.1 Evaluation setup

We use F-scores of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4 as evaluation metrics, which are computed based on the Chinese word segmentation results. Our proposed methods include **SummaryRank** and **EnsembleRank**. SummaryRank directly makes use of Ranking SVM for candidate summary ranking, as described in Sect. 5.1, while EnsembleRank makes use of our proposed top- $K$  ensemble ranking method to re-rank the candidate summaries. All features are used by SummaryRank and EnsembleRank. For EnsembleRank,  $K$  is tuned on the training set and set to 10, and  $P$  is simply set to 10. We use the following baseline methods for comparison:

*Baseline(EN)* It was introduced in Yao et al. [27] and relied merely on the English-side information for English sentence ranking in the original documents. The English summary is automatically translated into Chinese.

**Table 4** Word-level features

Feature	Description
TF	The sum of the word frequency in a summary, where word frequency is computed from the original document set
TFIDF	The sum of $TF_i * IDF_i$ in a summary, where $TF_i$ stands for the frequency of word $i$ in the summary. $IDF_i = \log \frac{ D }{ di }$ , where $ D $ stands for the total number of the document in the original document set and $ di $ stands for the number of the documents that contain word $i$
Stopword	The ratio of the stopwords number to the summary length
Unique word	The ratio of the unique word (i.e., word type) number to the summary length
Lead word	The ratio of the number of words that belong to the leading sentences of the document to the summary length
Word count	The number of words in the summary
Punc	The ratio of the number of punctuation marks to the summary length
POS	The ratio of the number of nouns, verbs and adjectives to the summary length
Named entity	The ratio of the number of named entities in a summary to the summary length

**Table 5** Sentence-level features

Feature	Description
Length	The mean/max/min length of the sentences in the summary
Position	The mean position weight of the sentences in the summary, where the weight of a sentence is calculated as $1 - \frac{(position-1)}{sentence\ count-1}$ . <i>Position</i> stands for the sentence position in the original document, while the sentence count stands for the total sentence number in the original document
Similarity with First Sentence	The mean cosine similarity between the sentences in the summary and the leading sentences in the document set
Sentence PageRank	The mean PageRank value of the sentences in the summary, where the nodes in the PageRank algorithm represent sentences in a given summary

**Baseline(CN)** It was introduced in Yao et al. [27] and relied merely on the Chinese-side information for Chinese sentence ranking in the translated Chinese documents.

**CoRank** We implement the graph-based CoRank algorithm proposed in Wan [22]. The algorithm makes use of bilingual information for cross-language summary extraction.

**PBES** It was proposed in Yao et al. [27] and made use of a phrase-based model for Chinese summary extraction. The score function relies on phrase-based alignment results.

**PBCS** It was also proposed in Yao et al. [27] which is an extension of PBES by performing sentence selection and sentence compression simultaneously. PBCS is the state-of-the-art unsupervised methods for cross-language summarization.

**Best submodular model** It refers to the best single model out of the 330 options proposed in the candidate summary extraction step. The model is determined based on its average performance over the training set and then applied on each document set in the test set.

**Table 6** Summary-level features

Feature	Description
Doc-Sum Cosine	The cosine similarity between the concatenated text of the original document set and the summary
Sum-Sum Cosine	The average cosine similarity between the summary and other candidate summaries
Doc-Sum Jaccard	The Jaccard similarity between the concatenated text of the original document set and the summary
Sum-Sum Jaccard	The average Jaccard similarity between the summary and other candidate summaries
Doc-Sum Dice	The Dice similarity between the concatenated text of the original document set and the summary
Sum-Sum Dice	The average Dice similarity between the summary and other candidate summaries
Max-Cos	The max cosine similarity between any two sentences in the summary
Mean-Cos	The mean cosine similarity between every pair of sentences in the summary
Sum PageRank	The PageRank value of the summary where the nodes in the PageRank algorithm represent candidate summaries

**Table 7** Readability-related features

Feature	Description
Conjunction	The sum of the weight of the non-juxtaposed conjunction word in the summary, where the weight of the conjunction $c$ is computed as $w(c) = 1 - \frac{\text{sentence}(c)-1}{\text{sentence count}-1}$ . $\text{Sentence}(c)$ stands for the position of the sentence that contains word $c$ in the summary and $\text{sentence count}$ stands for the total sentence number in the summary
Pronoun	The ratio of the number of pronouns in a summary to the summary length
Continuity	The mean continuity weight of the sentence in the summary. The continuity weight of sentence with the number $i$ is computed as: $C(i) = \cos(S_i, S_{i+1}) * \left(1 - \frac{i-1}{\text{sentence count}-1}\right)$ . $\cos(S_i, S_{i+1})$ stands for the cosine similarity between sentence $i$ and sentence $i+1$ , while $\text{sentence count}$ stands for the total sentence number in the summary
Bigram	The max/min/mean bigram probability of the sentence in the summary, where the probabilities of $P(\text{word1} \text{word2})$ are calculated from the Chinese-simplified dataset of the Google Ngrams Viewer

**SentenceRank** This is a supervised baseline based on supervised sentence ranking, as in Ouyang et al. [17]. Different from summary ranking, this baseline adopts Ranking SVM to rank sentences in the translated Chinese documents and then select sentences into summary in a greedy way. The target value of each sentence is measured as the maximum ROUGE score between the sentence and any sentence in the reference summaries. The features for each sentence include the sum of TFIDF scores, sentence length, sentence position, the average cosine similarity with other sentences, the similarity with the first sentence in the document, the similarity with neighboring sentences, and number of punctuation marks.

**SentenceSVR** This method is another supervised baseline based on supervised sentence regression, as in Ouyang et al. [17]. It adopts support vector regression to predict the score

**Table 8** Comparison results on test set

Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
BaselineEN (Google)	0.21684	0.03181	0.11550	0.06357
BaselineCN (Google)	0.23123	0.03919	0.12572	0.07111
CoRank (Google)	0.22859	0.04060	0.12721	0.07161
BaselineEN (Baidu)	0.22194	0.03177	0.12144	0.06600
BaselineCN (Baidu)	0.22942	0.03529	0.12552	0.07107
CoRank (Baidu)	0.23306	0.03726	0.13006	0.07447
BaselineEN (Youdao)	0.22975	0.03525	0.12135	0.06824
BaselineCN (Youdao)	0.22611	0.03869	0.12258	0.06927
CoRank (Youdao)	0.22950	0.04032	0.12672	0.07221
PBES	0.22825	0.04037	0.12856	0.06894
PBCS	0.24917	0.04632	0.13591	0.07953
Best submodular model	0.26709	0.04232	0.14374	0.08315
SentenceRank (Google)	0.26127	0.04455	0.14308	0.08361
SentenceSVR (Google)	0.26478	0.04506	0.14745	0.08426
SentenceRank (Baidu)	0.26091	0.04332	0.14216	0.08369
SentenceSVR (Baidu)	0.26102	0.04356	0.14194	0.08333
SentenceRank (Youdao)	0.26085	0.04642	0.14564	0.08340
SentenceSVR (Youdao)	0.27135	0.04590	0.14455	0.08458
SummaryRank	0.28492	0.04738	0.14755	0.08787
EnsembleRank	0.29250	0.04861	0.15314	0.09153

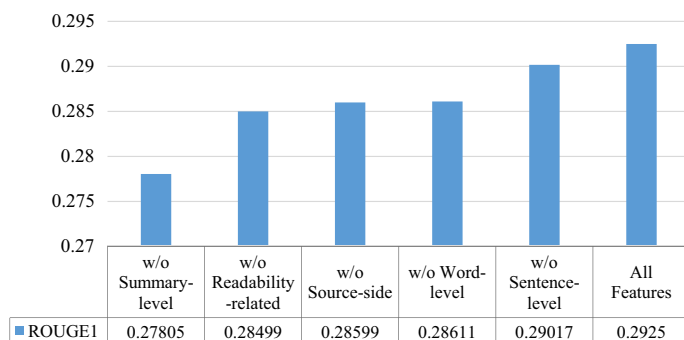
of each sentence. The target value and the features of each sentence are the same as that for SentenceRank.

Note that all the above baselines rely on a machine translation service. PBES and PBCS rely on bilingual phrase alignment results, which unfortunately could not be obtained from either Baidu Translate or Youdao Translation. So we directly use the performance values of PBES and PBCS reported in Yao et al. [27], which were based on Google Translate.

## 6.2 Comparison results

Table 8 shows the comparison results. The ROUGE scores for the baselines with different machine translation services are presented. We can see that our proposed SummaryRank and EnsembleRank methods outperform all baselines including the supervised SentenceRank and SentenceSVR methods, which demonstrates the effectiveness of our proposed framework for cross-language document summarization. More importantly, EnsembleRank outperforms SummaryRank, which further demonstrates the efficiency of our proposed ensemble ranking method.

As expected, the supervised baselines outperform most unsupervised baselines. The best submodular model outperforms other unsupervised baselines, while its performance is still much lower than our proposed EnsembleRank method. Regarding the three machine translation services, there is no clear evidence that one machine translation service is consistently better than the other two for all methods, although they do have impact on the performance of each method.



**Fig. 2** Feature validation for EnsembleRank

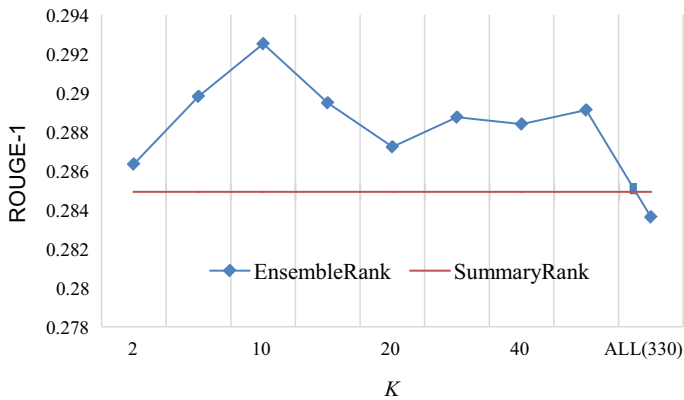
**Table 9**  $p$  Values of paired  $t$  tests between EnsembleRank and baseline methods

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
Best submodular model	0.00160	0.01310	0.01620	0.00318
SentenceRank (Google)	0.01741	0.07431	0.01926	0.04711
SentenceSVR (Google)	0.01380	0.04300	0.04970	0.12900
SentenceRank (Baidu)	0.00362	0.08370	0.00304	0.35200
SentenceSVR (Baidu)	0.00204	0.03460	0.03710	0.02480
SentenceRank (Youdao)	0.00672	0.26490	0.02920	0.06950
SentenceSVR (Youdao)	0.00527	0.06770	0.03770	0.05440
SummaryRank	0.04180	0.40100	0.04430	0.41480

We further perform paired  $t$  tests to show whether the performance difference is statistically significant. Table 9 shows the  $p$  values of paired  $t$  tests between our proposed method (EnsembleRank) and several strong baseline methods implemented by ourselves. We can see from the table that the ROUGE-1 or ROUGE-L performance improvement of EnsembleRank over baseline methods is statistically significant with  $p$  values lower than 0.05, while for the ROUGE-2 and ROUGE-SU4 scores, EnsembleRank can significantly outperform only some of the baseline methods. As mentioned in Sect. 4.2, we focus more on ROUGE-1 since the ROUGE scores are calculated based on Chinese words, and a Chinese word often consists of two or more Chinese characters. It is noteworthy that the performance difference between our proposed method and other baseline methods not listed in Table 9 is remarkably large and thus we believe that the performance improvement is statistically significant in most cases, especially with respect to ROUGE-1 and ROUGE-L.

We now investigate the influence of each group of features on the summarization performance. Figure 2 compares the ROUGE-1 scores of EnsembleRank with all features and after removing each group of features. We can see that the ROUGE score declines if any group of features is removed, which validates the usefulness of each group of features.

Figure 3 shows the influences of different  $K$  on the ROUGE-1 scores of our proposed EnsembleRank method. We also add the score of SummaryRank for comparison in the figure. We can see that the performance slightly fluctuates with different  $K$ . However, EnsembleRank can always outperform SummaryRank when  $K$  is set to a value in a wide range. The results demonstrate the robustness of our proposed ensemble ranking method. Note that when  $K$  is



**Fig. 3** The influence of  $K$  for EnsembleRank

**Table 10** EnsembleRank versus AverageRank

Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
AverageRank	0.28655	0.04742	0.14777	0.08863
SummaryRank	0.28492	0.04738	0.14755	0.08787
EnsembleRank	0.29250* <sup>#</sup>	0.04861	0.15314* <sup>#</sup>	0.09153

\* and <sup>#</sup> indicate that the performance improvement over AverageRank and SummaryRank is statistically significant, respectively:  $p$  value < 0.05 for paired  $t$  test

**Table 11** Randomly composed 330 candidates versus our 330 candidates

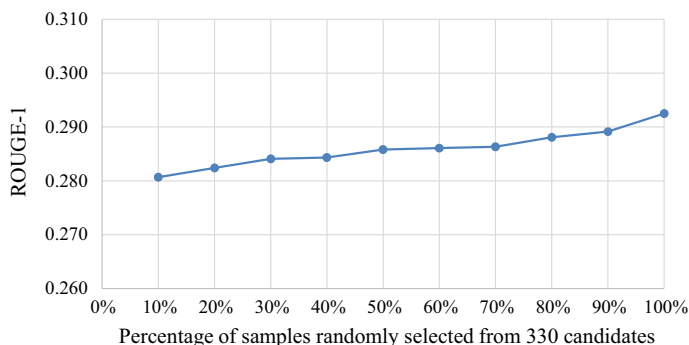
Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
Randomly composed candidates	0.26997	0.04193	0.14596	0.08366
Our candidates	0.29250*	0.04861*	0.15314*	0.09153*

\* Indicates that the performance improvement over randomly composed candidates is statistically significant:  $p$  value < 0.05 for paired  $t$  test

set to a very large value (e.g., 330), the performance of EnsembleRank declines sharply, due to the influences of the bad examples in the set.

We further compare our ensemble ranking method with a very simple ensemble method by averaging the 10 ranking positions for each candidate summary as its final position (denoted as AverageRank), as shown in Table 10. We can see that our proposed ensemble ranking method by learning a re-ranker from best ones performs better than the simple ensemble method.

In order to show the necessity of our candidate summary extraction step, we compare the performance of EnsembleRank over our extracted 330 candidate summaries and that over 330 randomly composed candidate summaries in Table 11. We can see that the performance over the randomly composed candidate summaries is not good, because the randomly composed set may not contain good summaries for each document set. We further investigate the influence of the number of candidate summaries on the performance of EnsembleRank. We randomly sample a portion of candidate summaries from the total 330 candidates and the sampling



**Fig. 4** The influence of number of candidate summaries on EnsembleRank

**Table 12** Manual evaluation results

	GR	NR	RC	TF	SC
CoRank	3.02 ± 0.74	3.22 ± 0.85	3.18 ± 0.63	3.31 ± 0.59	3.18 ± 0.76
SentenceRank	2.84 ± 0.59	2.93 ± 0.69	<b>3.31</b> ± 0.42	3.40 ± 0.59	3.01 ± 0.74
SentenceSVR	2.88 ± 0.64	3.12 ± 0.72	3.04 ± 0.53	3.24 ± 0.66	3.28 ± 0.67
SummaryRank	3.11 ± 0.81	3.65 ± 0.48	3.11 ± 0.81	<b>3.61</b> ± 0.52	3.22 ± 0.80
EnsembleRank	<b>3.17</b> ± 0.84	<b>3.72</b> ± 0.45	3.18 ± 0.85	3.55 ± 0.53	<b>3.29</b> ± 0.73

Bold indicates the highest values in each column

percentage ranges from 10 to 100%, and the performance curve is shown in Fig. 4. We can see that the performance declines with the decrease of the number of candidate summaries, but the decline trend is gentle. Whenever 10% candidate summaries are removed, the performance change is very slight. It is promising that when only 50% candidate summaries (i.e., 165 candidate summaries) are used for summary ranking, the performance of EnsembleRank can still be better than the strongest baseline—SummaryRank. In all, we can make use of a relatively small number of candidate summaries in our proposed method and it can still achieve promising performance.

Lastly, we employ two native Chinese students to manually evaluate the summaries produced by our methods and several typical baselines in five aspects. The aspects considered during evaluation include grammaticality (GR), non-redundancy (NR), referential clarity (RC), topical focus (TF), and structural coherence (SC). Each aspect is rated with scores from 1 (poor) to 5 (good). This evaluation is performed on the same random sample of 15 document sets from the test set. The average score and standard deviation for each metric are displayed in Table 12. We can see that our proposed methods bring certain amount of improvements on grammaticality and non-redundancy and also bring slight improvements of topic focus and structural coherence. The reason lies in that we make use of different features to characterize different aspects of a summary.

## 7 Related work

Multi-document summarization has been extensively investigated in the monolingual environment. Typical extraction-based multi-document summarization methods include the

centroid-based method [20], integer linear programming (ILP) [9], submodular function maximization [13–15], graph-based methods [5, 24], and supervised learning-based methods [4, 17, 18, 21]. Most recently, there are two studies trying to rank candidate summaries for multi-document summarization. Wan et al. [26] and Hong et al. [10] proposed to first produce a large number of candidate summaries by using different summarization systems and then learn to rank the summaries with a few off-the-shelf features. Different from Wan et al. [26] and Hong et al. [10], our study is performed in the cross-language environment, and the errors and noises brought by machine translation services make our study more challenging. In order to address the challenges, we propose and develop several unique techniques: (1) we have investigated more interesting strategies for candidate summary extraction from bilingual perspectives; (2) we derive a few new features for ranking translated candidate Chinese summaries, including readability-related features and features from both languages; (3) more importantly, we propose a new ensemble ranking method as a re-ranker to aggregate the ranking results of the basic rankers. Based on these techniques, our proposed method achieves state-of-the-art performance on the cross-language summarization task.

The task of cross-language summarization has not been well studied. There are only a few pilot studies focusing on this task [16, 19, 22, 23, 27]. All these studies rely on sentence evaluation and selection in the target language, and the overall performance is not satisfactory.

## 8 Conclusion and future work

In this paper, we proposed a new framework for addressing the cross-language document summarization task by extraction and ranking of multiple summaries in the target language. The top- $K$  ensemble ranking method is proposed for ranking candidate summaries. Evaluation results on a benchmark dataset validate the effectiveness of our proposed framework and the ensemble ranking method.

In future work, we will test the robustness of our proposed framework in other target languages, e.g., Japanese, French. We will also try to use deep learning techniques for learning latent features to improve summary ranking. Lastly, we will explore different summarization methods to produce more diversified candidate summaries for ranking, and we believe the cross-language document summarization performance will be further improved.

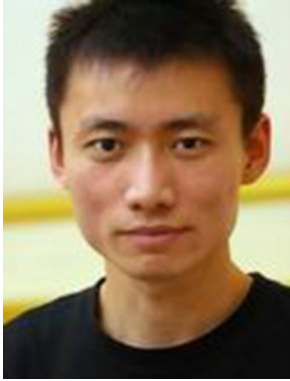
**Acknowledgements** This work was supported by National Natural Science Foundation of China (61331011, 61772036), IBM Global Faculty Award Program, and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

## References

1. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on machine learning. pp 89–96
2. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
3. Cao Z, Qin T, Liu T-Y, Tsai M-F, Li H (2007) Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th international conference on machine learning. pp 129–136
4. Cao Z, Wei F, Dong L, Li S, Zhou M (2015) Ranking with recursive neural networks and its application to multi-document summarization. In: Proceedings of AAAI. pp 2153–2159
5. Erkan G, Radev D (2004) LexPageRank: Prestige in multi-document text summarization. In: Proceedings of EMNLP. pp 365–371



6. Filippova K (2010) Multi-sentence compression: finding shortest paths in word graphs. In: Proceedings of the 23rd international conference on computational linguistics (Coling 2010). pp 322–330
7. Freund Y, Iyer R, Schapire RE, Singer Y (2003) An efficient boosting algorithm for combining preferences. *J Mach Learn Res* 4:933–969
8. Frey BJ, Dueck D (2007) Clustering by passing messages between data points. *Science* 315(5814):972–976
9. Gillick D, Favre B, Hakkani-Tur D (2008) The ICSI summarization system at TAC 2008. In: Proceedings of the text understanding conference
10. Hong K, Marcus M, Nenkova A (2015) System combination for multi-document summarization. In: Proceedings of EMNLP. pp 107–117
11. Joachims T (2002) Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. pp 133–142
12. Lin CY (2004) Rouge: a package for automatic evaluation of summaries. In: Proceedings of the ACL-04 workshop on text summarization branches out
13. Li J, Li L, Li T (2012) Multi-document summarization via submodularity. *Appl Intell* 37(3):420–430
14. Lin H, Bilmes J (2010) Multi-document summarization via budgeted maximization of submodular functions. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, pp 912–920
15. Lin H, Bilmes J (2011) A class of submodular functions for document summarization. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies-volume 1, Association for computational linguistics. pp 510–520
16. Orasan C, Chiorean OA (2008) Evaluation of a cross-lingual romanian-english multi-document summariser. In: Proceedings of LREC
17. Ouyang Y, Li W, Li S, Lu Q (2011) Applying regression models to query-focused multi-document summarization. *Inf Process Manag* 47:227–237
18. Ouyang Y, Li S, Li W (2007) Developing learning strategies for topic-based summarization. In: Proceedings of the Sixteenth ACM conference on information and knowledge management, ACM. pp 79–86
19. Pingali P, Jagarlamudi J, Varma V (2007) Experiments in cross language query focused multi-document summarization. In: Workshop on cross lingual information access addressing the information need of multilingual societies in IJCAI2007
20. Radev D, Jing H, Styś M, Tam D (2004) Centroid-based summarization of multiple documents. *Inf Process Manag* 40(6):919–938
21. Shen D, Sun JT, Li H, Yang Q, Chen Z (2007) Document summarization using conditional random fields. In: Proceedings of IJCAI. pp 2862–2867
22. Wan X (2011) Using bilingual information for cross-language document summarization. In: Proceedings of ACL. pp 1546–1555
23. Wan X, Li H, Xiao J (2010) Cross-language document summarization based on machine translation quality prediction. In: Proceedings of ACL. pp 917–926
24. Wan X, Yang J, Xiao J (2007) Manifold-ranking based topic-focused multi-document summarization. In: Proceedings of IJCAI. pp 2903–2908
25. Wan X, Yang J (2008) Multi-document summarization using cluster-based link analysis. In: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval. pp 299–306
26. Wan X, Cao Z, Wei F, Li S, Zhou M (2015) Multi-document summarization via discriminative summary reranking. [arXiv:1507.02062](https://arxiv.org/abs/1507.02062)
27. Yao JG, Wan X, Xiao J (2015) Phrase-based compressive cross-language summarization. In: Proceedings of EMNLP. pp 118–127



**Xiaojun Wan** is a professor at Institute of Computer Science and Technology of Peking University. He is also affiliated with Key Laboratory of Computational Linguistics (Peking University), MOE. He received his B. S., M. S. and Ph. D degrees from Peking University in 2000, 2003 and 2006, respectively. His research interests include natural language processing and text mining. He has served as PC member or area chair of major conferences such as ACL, SIGIR, WWW, AAAI, IJCAI, EMNLP, CIKM, and IJCNLP.



**Fuli Luo** was an intern student at Institute of Computer Science and Technology of Peking University when she conducted this research. She is currently a master student at Key Laboratory of Computational Linguistics (Peking University), MOE. His research interests include natural language processing.



**Xue Sun** is an intern student at Institute of Computer Science and Technology of Peking University. She is currently a master student at Department of Computer and Information Science, UCLA. Her research interests include natural language processing.



**Songfang Huang** is a Research Staff Member in IBM China Research Lab. Before joining CRL in 2011, he was a postdoctoral researcher of IBM Watson Research Lab, working on Speech-to-Speech translation. He obtained his PhD degree from University of Edinburgh, UK in 2009.



**Jin-ge Yao** was a Ph.D. student at Institute of Computer Science and Technology of Peking University when he conducted this research. He is currently a researcher in Microsoft Research Asia. His research interests include natural language processing.