Automatic Generation of Related Work Sections in Scientific Papers: An Optimization Approach

Yue Hu and Xiaojun Wan
Institute of Computer Science and Technology
The MOE Key Laboratory of Computational Linguistics
Peking University, Beijing, China
{ayue.hu,wanxiaojun}@pku.edu.cn

Abstract

In this paper, we investigate a challenging task of automatic related work generation. Given multiple reference papers as input, the task aims to generate a related work section for a target paper. The generated related work section can be used as a draft for the author to complete his or her final related work section. We propose our Automatic Related Work Generation system called ARWG to address this task. It first exploits a PLSA model to split the sentence set of the given papers into different topic-biased parts, and then applies regression models to learn the importance of the sentences. At last it employs an optimization framework to generate the related work section. Our evaluation results on a test set of 150 target papers along with their reference papers show that our proposed ARWG system can generate related work sections with better quality. A user study is also performed to show ARWG can achieve an improvement over generic multi-document summarization baselines.

1 Introduction

The related work section is an important part of a paper. An author often needs to help readers to understand the context of his or her research problem and compare his or her current work with previous works. A related work section is often used for this purpose to show the differences and advantages of his or her work, compared with related research works. In this study, we attempt to automatically generate a related work section for a target academic paper with its reference papers. This kind of related work sections can be used as a basis to reduce the author's time and effort when he or she wants to complete his or her final related work section.

Automatic related work section generation is a very challenging task. It can be considered a topic-biased, multiple-document summarization problem. The input is a target academic paper, which has no related work section, along with its reference papers. The goal is to create a related work section that describes the related works and addresses the relationship between the target paper and the reference papers. Here we assume that the set of reference papers has been given as part of the input. Existing works in the NLP and recommendation systems communities have already focused on the task of finding reference papers. For example, citation prediction (Nallapati et al., 2008) aims at finding individual paper citation patterns.

Generally speaking, automatic related work section generation is a strikingly different problem and it is much more difficult in comparison with general multi-document summarization tasks. For example, multi-document summarization of news articles aims at synthesizing contents of similar news and removing the redundant information contained by the different news articles. However, each scientific paper has much specific content to state its own work and contribution. Even for the papers that investigate the same research topic, their contributions and contents can be totally different. The related work section generation task needs to find the specific contributions of individual papers and arrange them into one or several paragraphs.

In this study, we focus on the problem of automatic related work section generation and propose a novel system called ARWG to address the

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1624–1633,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

problem. For the target paper, we assume that the abstract and introduction sections have already been written by the author and they can be used to help generate the related work section. For the reference papers, we only consider and extract the abstract, introduction, related work and conclusion sections, because other sections like the method and evaluation sections always describe the extreme details of the specific work and they are not suitable for this task. Then we generate the related work section using both sentence sets which are extracted from the target paper and reference papers, respectively.

Firstly, we use a PLSA model to group both sentence sets of the target paper and its reference papers into different topic-biased clusters. Secondly, the importance of each sentence in the target paper and the reference papers is learned by using two different Support Vector Regression (SVR) models. At last, a global optimization framework is proposed to generate the related work section by selecting sentences from both the target paper and the reference papers. Meanwhile, the framework selects sentences from different topic-biased clusters globally.

Experimental results on a test set of 150 target papers show our method can generate related work sections with better quality than those of several baseline methods. With the ROUGE toolkit, the results indicate the related work sections generated by our system can get higher ROUGE scores. Moreover, our related work sections can get higher rating scores based on a user study. Therefore, our related work sections can be much more suitable for the authors to prepare their final related work sections.

2 Related Work

There are few studies to directly address automatic related work generation. Hoang and Kan (2010) proposed a related work summarization system given the set of keywords arranged in a hierarchical fashion that describes the paper's topic. They used two different rule-based strategies to extract sentences for general topics as well as detailed ones.

A few studies focus on multi-document scientific article summarization. Agarwal et al., (2011) introduced an unsupervised approach to the problem of multi-document summarization. The input is a list of papers cited together within the same source article. The key point of this approach is a topic based clustering of fragments extracted from each co-cited article. They rank all the clusters using a query generated from the context surrounding the co-cited list of papers. Yeloglu et al., (2011) compared four different approaches for multi-document scientific articles summarization: MEAD, MEAD with corpus specific vocabulary, LexRank and W3SS.

Other studies investigate mainly on the single-document scientific article summarization. Early works including (Luhn 1958; Baxendale 1958; Edumundson 1969) tried to use various features specific to scientific text (e.g., sentence position, or rhetorical clues features). They have proved that these features are effective for the scientific article summarization. Citation information has been already shown effective in summarize the scientific articles. Works including (Mei and Zhai 2008; Qazvinian and Radev 2008; Schwartz and Hearst 2006; Mohammad et al., 2009) employed citation information for the single scientific article summarization. Earlier work (Nakov et al., 2004) indicated that citation sentences may contain important concepts that can give useful descriptions of a paper.

Various methods have been proposed for news document summarization, including rule-based methods (Barzilay and Elhadad 1997; Marcu and Daniel 1997), graph-based methods (Mani and Bloedorn 2000; Erkan and Radev 2004; Michalcea and Tarau 2005), learning-based methods (Conroy et al., 2001; Shen et al., 2007; Ouyang et al., 2007; Galanis et al., 2008), optimization-based methods (McDonald 2007; Gillick et al., 2009; Xie et al., 2009; Berg-Kirkpatrick et al., 2011; Lei Huang et al., 2011; Woodsend et al., 2012; Galanis 2012), etc.

The most relevant work is (Hoang and Kan, 2010) as mentioned above. They also assumed the set of reference papers was given as part of the input. They also adopt the hierarchical topic tree that describes the topic structure in the target paper as an essential input for their system. However, it is non-trivial to build the hierarchical topic tree. Moreover, they do not consider the content of the target paper to construct the related work section, which is actually crucial in the related work section. To the best of our knowledge, no previous works have used supervised learning and optimization framework to deal with the multiple scientific article summarization tasks.

3 Problem Analysis and Corpus

3.1 Problem Analysis

We firstly analyze the structure of related work sections briefly. By using examples for illustration, we can gain insight on how to generate related work sections. A specific related work example is shown in Figure 1.

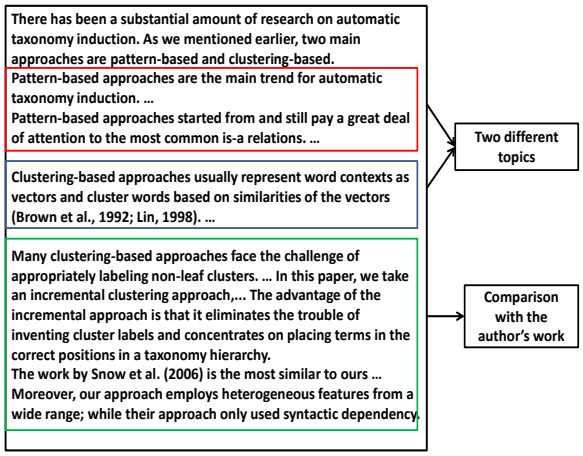

Figure 1: A sample related work section (Yang and Callan 2009)

This related work section introduces previous related works for a paper on Automatic Taxonomy Induction. From Figure 1, we can have a glance at the structure of related work sections. Related work sections usually discuss several different topics, such as "pattern-based" and "cluster-based" approaches shown in the Figure 1. Besides the knowledge of previous works, the author often compares his own work with the previous works. The differences and advantages are generally mentioned. The example in Figure 1 also indicates this phenomenon.

Therefore, we design our system to generate related work sections according to the related work section structure mentioned above. Our system takes the target paper for which a related work section needs to be drafted besides its reference papers as input. The goal of our system is to generate a related work section with the above structure. The generated related work section should have several topic-biased parts. The author's own work is also needed to be described and its difference with other works is needed to be emphasized on.

3.2 Corpus and Preprocessing

We build a corpus that contains academic papers and their corresponding reference papers. The academic papers are selected from the ACL Anthology[1]. The ACL Anthology currently hosts over 24,500 papers from major conferences such as ACL, EMNLP, COLING in the fields of computational linguistics and natural language processing. We remove the papers that contain related work sections with very short length, and randomly select 1050 target papers to construct our whole corpus.

The papers are all in PDF format. We extract their texts by using PDFlib[2] and detect their physical structures of paragraphs, subsections and sections by using ParsCit[3]. For the target papers, the related work sections are directly extracted as the gold summaries. The references are also extracted. For the references that can be found in the ACL Anthology, we download them from the ACL Anthology. The other reference papers are searched and downloaded by using Google Scholar. References to books and PhD theses are discarded, for their verbosity may change the problem drastically (Mihalcea and Ceylan, 2007).

The input of our system includes the abstract and introduction sections of the target paper, and the abstract, introduction, related work and conclusion sections of the reference papers. As mentioned above, the method and evaluation sections in the reference papers are not used as input because these sections usually describe extreme details of the methods and evaluation results and they are not suitable for related work generation. Note that it is reasonable to make use of the abstract and introduction sections of the target paper to help generate the related work section, because an author usually has already written the abstract and introduction sections before he or she wants to write the related work section for the target paper. Otherwise, we cannot get any information about the author's own work. All other sections in the target paper are not used.

4 Our Proposed System

4.1 Overview

In this paper, we propose a system called ARWG to automatically generate a related work section for a given target paper. The architecture of our system is shown in Figure 2. We take both the target paper and its reference papers as input and they are represented by several sections mentioned in Section 3.2. After preprocessing, we extract the feature vectors for sentences in the target paper and the reference papers, respective-

[1] http://aclweb.org/anthology/

[2] http://www.pdflib.com/

[3] http://aye.comp.nus.edu.sg/parsCit/

ly. The importance scores for sentences in the target paper and the reference papers are assigned by using two SVR based sentence scoring models. The two SVR models are trained for sentences in the target paper and the reference papers, respectively. Meanwhile, a topic model is applied to the whole set of sentences in both the target paper and reference papers. The sentences are grouped into several different topic-biased clusters. The sentences with importance scores and topic cluster information are taken as the input for the global optimization framework. The optimization framework extracts sentences to describe both the author's own work and background knowledge. More details of each part will be discussed in the following sections.

4.2 Topic Model Learning

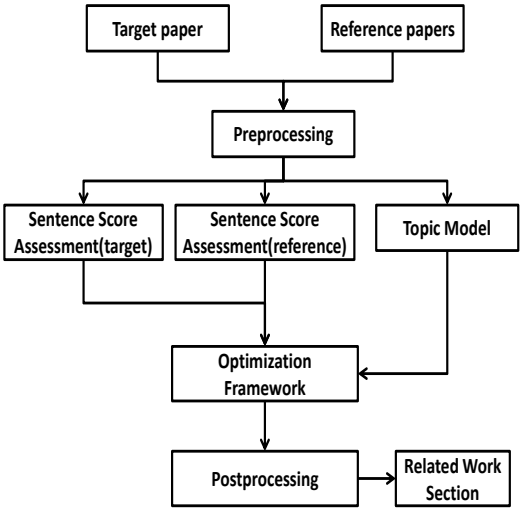

Figure 2: System Architecture

As mentioned in the previous section, the related work section usually addresses several different topics. The topics may be different research themes or different aspects of a broad research theme. The related work section should describe the specific details for each topic, respectively.

Therefore, we aim to discover the hidden topics of the input papers, and we use the Probabilistic latent semantic analysis (PLSA) (Hofmann, 1999) to solve this problem.

The PLSA approach models each word in a document as a sample from a mixture model. The mixture components are multinomial random variables that can be viewed as representations of "topics". Different words in a document may be generated from different topics. Each document is represented a list of mixing proportions for these mixture components and can be reduced to a probability distribution on a fixed set of topics.

Considering that the sentences in one paper may relate to different topics, we treat each sentence as a "document" *d*. We treat the noun phases in the sentences as the "words" *w*. In order to extract the noun phrases, chunking implemented by the OpenNLP toolkit [4] is applied to the sentences. Noun phrases that contain words such as "paper" and "data" are discarded.

Then the sentences with their corresponding noun phrases are taken as input into the PLSA model. Here both the sentences in the target paper and the sentences in the reference papers are treated the same in the model. Finally, we can get the sentence set with topic information and use it in the subsequent steps. Each sentence has a topic weight *t* in each topic.

4.3 Sentence Important Assessment

In our proposed system, sentence importance assessment aims to assign an importance score to each sentence in the target paper and reference papers. The score of each sentence will be used in the subsequent optimization framework. We propose to use the support vector regression model to achieve this goal. In the above topic model learning process, we do not distinguish the sentences in the target paper and reference papers. In contrast, we train two different support vector regression models separately for the sentences in the target paper and the sentences in the reference papers. In the related work section, the sentences that describe the author's own work usually address the differences from the related works, while the sentences that describe the related works often focus on the specific details. We think the two kinds of sentences should be treated differently.

Scoring Method

To construct training data based on the papers collected, we apply a similarity scoring method to assign the importance scores to the sentences in the papers. The main hypothesis is that the sentences in the gold related work sections should summarize the target paper and reference papers as well. Thus the sentences in the papers which are more similar to the sentences in the gold related work sections should be considered more important and suitable to be selected. Our scoring method should assign higher scores to them.

[4] http://opennlp.apache.org/

We define the importance score of a sentence in the papers as below:

$$score(s) = \max_{s_i^* \in S^*}(sim(s, s_i^*)) \quad (1)$$

where s is a sentence in the papers, S^* is the set of the sentences in the corresponding gold related work section. The standard cosine measure is employed as the similarity function.

Considering the difference between the sentences that describe the author's work and the sentences that describe the related works, we split the set of sentences in the gold related work section into two parts: one discusses the author's own work and the other introduces the related works. We observe that sentences related to the author's own work often feature specific words or phrases (such as "we", "our work", "in this paper" etc.) in the related work section. So we check the sentences about whether they contain clue words or phrases (i.e., "in this paper", "our work" and 18 other phrases). If the clue phrase check fails, the sentence belongs to the related work part. If not, it belongs the own work part.

Thus for the sentences in the target paper, S^* is the set of sentences in the own work part of the gold related work section, while for the sentences in the reference papers, S^* is the set of sentences in the related work part of the gold related work section. Then we can use the scoring method to compute the target scores of the sentences in the training set. It is noteworthy that two SVR models can be trained on the two parts of the training data, respectively.

Feature

Each sentence is represented by a set of features. The common features used for the sentences of the target paper and reference papers are shown in Table 1. The additional features applied to the sentences of the target paper are introduced in Table 2.

Here, *s* is a sentence that needs to extract features. *th* is paper title, section headings and subsection headings set of the reference papers or target paper for the two SVR models, respectively. Each feature with "*" represent a feature set that contains similar features.

All the features are scaled into [0, 1]. Thus we can learn SVR models based on the features and importance scores of the sentences, and then use the models to predict an importance score for each sentence in the test set. The SVR models are trained and applied for the target paper and reference papers, respectively.

Table 1: Common features employed in the SVR models

Feature	Description
$Sim(s,th)^*$	The similarity between *s* and each title in *th*; Stop words are removed and stemming is employed.
WS(s,th)	Number of words shared by *s* and *th*.
$SP(s)^*$	The position of *s* in its section or subsection
$PTI(s)^*$	The parse tree information of *s*, including the number of noun phrase and verb phrases, the depth of the parse tree, etc.
$IsHead(s)^*$	Indicates whether *s* is the first sentence of the section or subsection
$IsEnd(s)^*$	Indicates whether *s* is the last sentence of the section or subsection
SWP(s)	The percentage of the stop words
Length(s)	The length of sentence *s*
Length_rw(s)	The length of *s* after removing stop words
SI(s)	The section index of *s* that indicates which section *s* is from.
$CluePhrase(s)^*$	Indicates whether a clue phrase appears in *s*. the clue phrases include "our work", "propose" and other 20 words. Each clue phrase corresponds to one feature.

Table 2: Additional features for sentences in the target paper

Feature	Description
HasCitation(s)	Indicates whether *s* contains a citation
$PhraseForCmp(s)^*$	Indicates whether *s* contains words or phrases used for comparison such as "in contrast", "instead" and other 26 words. Each word or phrase corresponds to one feature.

4.4 A Global Optimization Framework

In the above steps, we can get the predicted importance score and topic information for each sentence in the target paper and reference papers. Here, we introduce a global optimization framework to generate the related work section.

According to the structure of the related work section mentioned above, the related work section usually discusses several topics. In each topic, the related works and their details are introduced. Besides, the author often compares his own work with these previous works.

Therefore, we propose to formulate the generation as an optimization problem. Basically, we will be searching for a set of sentences to optimize the objective function.

Table 3: Notations used in this section

Symbol	Description
sr_i/st_i	the sentence in the reference/target paper
lr_i/lt_i	the length of sentence sr_i/ st_i
wr_i/wt_i	the importance score of sr_i/st_i
xr_{ij}/xt_{ij}	indicates whether sr_i/st_i is selected into the part of topic *j* in the generated related work section
nr/nt	the number of sentences in the reference/target papers
m	the topic count
t_{ij}	the topic weight of sr_i/st_i in topic j from the PLSA model
B	the set of unique bigrams
y_i	indicates whether bigram b_i is included in the result
c_{b_i}	the count of the occurrences of bigram b_i in the both target paper and reference papers
L_{max}	the maximum word count of the related work section
L_j	the maximum word count of the part of topic j which depends on the percentage of sentences belong to topic j
B^*	the total set of bigrams in the whole paper set
B_i	the set of bigrams that sentence sr_i/st_i contains
Sr_m/St_m	the set of sentences that include bigram b_m in the reference/target papers
$\lambda_1, \lambda_2, \lambda_3$	parameters for tuning

To design the objective function, three aspects should be considered:

1) First, the related work section we generate should introduce the previous works well. In our assumption, sentences with higher importance scores are better to be selected. In addition, very short sentences should be penalized. So we introduce the first part of our objective function below:

$$\sum_{i=1}^{nr}(lr_i wr_i \sum_{j=1}^{m} t_{ij} xr_{ij}) \quad (2)$$

We add the sentence length as a multiplication factor in order to penalize the very short sentences, or the objective function tends to select more and shorter sentences. At the same time, the objective function does not tend to select the very long sentences. The total length of the sentences selected is fixed. So if the objective function tends to select the longer sentences, the fewer sentences can be selected. A tradeoff needs to be made between the number and the average length of the sentences selected.

The constraints introduced below ensure that the sentence can only be selected into one topic and the topic weight is used to measure the degree that the sentence is relevant to the specific topic.

2) Second, similar to the first part, we should consider the own work part of the related work section. Thus the second part of our objective function is shown as follows:

$$\sum_{i=1}^{nt}(lt_i wt_i \sum_{j=1}^{m} t_{ij} xt_{ij}) \quad (3)$$

3) At last, redundancy reduction should be considered in the objective function. The last part of the objective function is shown below:

$$\sum_{i=1}^{|B|} c_{b_i} y_i \quad (4)$$

The intuition is that the more unique bigrams the related work section contains, the less redundancy the related work section has. We add c_{b_i} as the weight of the bigram in order to include more important bigrams.

By combing all the parts defined above, we have the following full objective function:

$$\max_{xr,xt} \lambda_1 \sum_{i=1}^{nr}(\frac{lr_i}{\alpha L_{max}} wr_i \sum_{j=1}^{m} t_{ij} xr_{ij}) + \lambda_2 \sum_{i=1}^{nt}(\frac{lt_i}{(1-\alpha)L_{max}} wt_i \sum_{j=1}^{m} t_{ij} xt_{ij}) + \lambda_3 \sum_{i=1}^{|B|} \frac{c_{b_i} y_i}{|B^*|} \quad (5)$$

Subject to:

$$\sum_{i=1}^{nr} lr_i xr_{ij} + \sum_{i=1}^{nt} lt_i xt_{ij} < L_j, for\ j = 1, \dots, m \quad (6)$$

$$\sum_{i=1}^{nr} \sum_{j=1}^{m} lr_i xr_{ij} < \alpha L_{max} \quad (7)$$

$$\sum_{i=1}^{nt} \sum_{j=1}^{m} lt_i xt_{ij} < (1-\alpha) L_{max} \quad (8)$$

$$\sum_{j=1}^{m} xr_{ij} \le 1, for\ i = 1, \dots, nr \quad (9)$$

$$\sum_{j=1}^{m} xt_{ij} \le 1, for\ i = 1, \dots, nt \quad (10)$$

$$\sum_{b_k \in B_i} y_k \ge |B_i| \sum_{j=1}^{m} xr_{ij}, for\ i = 1, \dots, nr \quad (11)$$

$$\sum_{b_k \in B_i} y_k \ge |B_i| \sum_{j=1}^{m} xt_{ij}, for\ i = 1, \dots, nt \quad (12)$$

$$\sum_{sr_i \in Sr_k} \sum_{j=1}^{m} xr_{ij} + \sum_{st_i \in St_k} \sum_{j=1}^{m} xt_{ij} \ge y_k, \quad k = 1, \dots |B| \quad (13)$$

$$xr_{ij}, xt_{ij}, y_i \in \{0,1\} \quad (14)$$

All the three parts in the objective function are normalized to [0, 1] by using the maximum length L_{max} and the total number of bigrams $|B^*|$. λ_1, λ_2 and λ_3 are parameters for tuning the three parts and we set $\lambda_1+\lambda_2+\lambda_3 = 1$.

We explain the constraints as follows:

Constraint (6): It ensures that the total word count of the part of topic *j* does not exceed L_j.

Constraints (7), (8): The two constraints try to balance the lengths of the previous works part and the own work part, respectively. α is set to 2/3.

Constraints (9), (10): These two constraints guarantee that the sentence can only be included into one topic.

Constraints (11), (12): When these two constraints hold, all bigrams that s_i has are selected if s_i is selected.

Constraint (13): This constraint makes sure that at least one sentence in Sr_m or St_m is selected if bigram b_m is selected.

Therefore, we transform our optimization problem into a linear programing problem. We solve this linear programming problem by using the IBM CPLEX optimizer[5]. It generally takes tens of seconds to solve the problem and it is very efficient.

Finally, ARWG post-processes sentences to improve readability, including replacing agentive forms with a citation to the specific article (e.g., "our work" → "(Hoang and Kan, 2010)") for the sentences extracted from reference papers. The sentences belonging to different topics are placed separately.

5 Evaluation

5.1 Evaluation Setup

To set up our experiments, we divide our dataset which contains 1050 target papers and their reference papers into two parts: 700 target papers for training, 150 papers for test and the other 200 papers for validation. The PLSA topic model is applied to the whole dataset. We train two SVR regression models based on the own work part and the previous work part of the training data and apply the models to the test data. The global optimization framework is used to generate the related work sections. We set the maximum word count of the generated related work section to be equal to that of the gold related work section. The parameter values of λ_1, λ_2 and λ_3 are set to 0.3, 0.1 and 0.6, respectively. The parameter values are tuned on the validation data.

We compare our system with five baseline systems: MEAD-WT, LexRank-WT, ARWG-WT, MEAD and LexRank. MEAD[6] (Radev et al., 2004) is an open-source extractive multi-document summarizer. LexRank[7] (Eran and Radev, 2004) is a multi-document summarization system which is based on a random walk on the similarity graph of sentences. We also implement the MEAD, LexRank baselines and our method with only the reference papers (i.e. the target paper's content is not considered). Those methods are signed by "-WT".

To evaluate the effectiveness of the SVR models we employ, we implement a baseline system RWGOF that uses the random walk scores as the important scores of the sentences and take the scores as inputs for the same global optimization framework as our system to generate the related work section. The random walk scores are computed for the sentences in the reference papers and the target paper, respectively.

We use the ROUGE toolkit to evaluate the content quality of the generated related work sections. ROUGE (Lin, 2004) is a widely used automatic summarization evaluation method based on n-gram comparison. Here, we use the F-Measure scores of ROUGE-1, ROUGE-2 and ROUGE-SU4. The model texts are set as the gold related work sections extracted from the target papers, and word stemming is utilized. ROUGE-N is an n-gram based measure between a candidate text and a reference text. The recall oriented score, the precision oriented score and the F-measure score for ROUGE-N are computed as follows:

$$ROUGE - N_{Recall} = \sum_{S\in\{Reference\ Text\}}\sum_{gram_n} Count_{match}(gram_n) \,/\, \sum_{S\in\{Reference\ Text\}}\sum_{gram_n} Count(gram_n) \quad (15)$$

$$ROUGE - N_{Precision} = \sum_{S\in\{Reference\ Text\}}\sum_{gram_n} Count_{match}(gram_n) \,/\, \sum_{S\in\{Candidate\ Text\}}\sum_{gram_n} Count(gram_n) \quad (16)$$

$$ROUGE - N_{F-measure} = 2 * ROUGE - N_{Recall} * ROUGE - N_{Precision} \,/\, ROUGE - N_{Recall} + ROUGE - N_{Precision} \quad (17)$$

where n stands for the length of the n-gram $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate text and a reference text.

In addition, we conducted a user study to subjectively evaluate the related work sections to get more evidences. We selected the related work sections generated by different methods for 15 random target papers in the test set. We asked three human judges to follow an evaluation guideline we design and evaluate these related work sections. The human judges are graduate students in the computer science field and they did not know the identities of the evaluated related work sections. They were asked to give a rating on a scale of 1 (very poor) to 5 (very good) for the correctness, readability and usefulness of the related work sections, respectively:

[5] www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[6] http://www.summarization.com/mead/

[7] In our experiments, LexRank performs much better than the more complex variant - C-LexRank (Qazvinian and Radev, 2008), and thus we choose LexRank, rather than C-LexRank, to represent graph-based summarization methods for comparison in this paper.

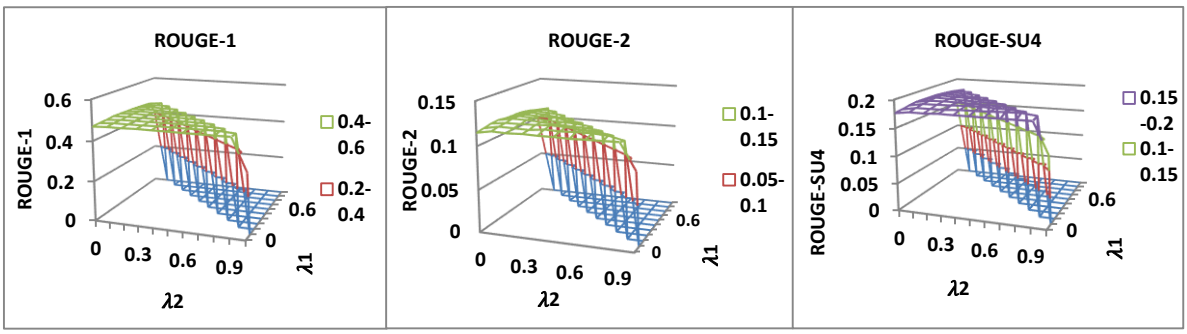

Figure 3: Parameter influences (horizontal, vertical axis are λ_1, λ_2 , respectively, $\lambda_3 = 1 - \lambda_1 - \lambda_2$)

1) Correctness: Is the related work section actually related to the target paper?
2) Readability: Is the related work section easy for the readers to read and grasp the key content?
3) Usefulness: Is the related work section useful for the author to prepare their final related work section?

Paired T-Tests are applied to both the ROUGE scores and rating scores for comparing ARWG and baselines and comparing the systems with WT and without WT.

5.2 Results and Discussion

Table 4: ROUGE F-measure comparison results

Method	ROUGE-1	ROUGE-2	ROUGE-SU4
Mead-WT	0.39720	0.08785	0.14694
LexRank-WT	0.43267	0.09228	0.16312
ARWG-WT	0.45077$^{*\{1,2\}}$	0.09987$^{*\{1,2\}}$	0.16731$^{*\{1\}\#\{2\}}$
Mead	0.41012$^{*\{1\}}$	0.09642$^{*\{1\}}$	0.15441$^{*\{1\}}$
LexRank	0.44235$^{*\{2\}}$	0.10090$^{*\{2\}}$	0.17067$^{*\{2\}}$
ARWG	**0.47940**$^{*\{1-5\}}$	**0.12176**$^{*\{1-5\}}$	**0.18618**$^{*\{1-5\}}$

(* represents pairwise t-test value $p < 0.01$; $^{\#}$ represents $p < 0.05$; the numbers in the brackets represent the indices of the methods compared, e.g. 1 for MEAD-WT, 2 for LexRank-WT, etc.)

Table 5: Average rating scores of judges

Method	Correctness	Readability	Usefulness
Mead	2.971	2.664	2.716
LexRank	2.958	2.847	2.784
ARWG	3.433$^{*\#}$	3.420$^{*\#}$	3.382$^{*\#}$

($^{*\#}$ represents pairwise t-test value $p < 0.01$, compared with Mead and LexRank, respectively.)

Table 6: ROUGE F-measure comparison of different sentence importance scores

Method	ROUGE-1	ROUGE-2	ROUGE-SU4
RWGOF	0.46932	0.11791	0.18426
ARWG	**0.47940**	**0.12176**	**0.18618**

The evaluation results over ROUGE metrics are presented in Table 4. It shows that our proposed system can get higher ROUGE scores, i.e., better content quality. In our system, we split the sentence set into different topic-biased parts, and the importance scores of sentences in the target paper and reference papers are learned differently. So the obtained importance scores of the sentences are more reliable.

The global optimization framework considers the extraction of both the previous work part and the own work part. We can see the importance of the own work part by comparing the results of the methods with or without considering the own work part. MEAD, LexRank and our method all get a significant improvement after considering the own work part by extracting sentences from the target paper. The results also prove our assumption about the related work section structure.

Figure 3 presents the fluctuation of ROUGE scores when tuning the parameters λ_1, λ_2 and λ_3. We can see our method generally performs better than the baselines. All the three parts in the objective function are useful to generate related work sections with good quality.

The average scores rated by human judges for each method are showed in Table 5. We can see that the related work sections generated by our system are more related to the target papers. Moreover, because of the good structure of our generated related work sections, our generated related work sections are considered more readable and more useful for the author to prepare the final related work sections.

T-test results show that the performance improvements of our method over baselines are statistically significant on both automatic and manual evaluations. Most of p-values for t-test are far smaller than 0.01.

Overall, the results indicate that our method can generate much better related work sections

than the baselines on both automatic and human evaluations.

Table 6 shows the comparison results between ARWG and RWGOF. We can see ARWG performs better than RWGOF. It proves that the SVR models can better estimate the importance scores of the sentences. For the SVR models are trained from the large dataset, the sentence scores predicted by the SVR models can be more reliable to be used in the global optimization framework.

6 Conclusion and Future Work

This paper proposes a novel system called ARWG to generate related work sections for academic papers. It first exploits a PLSA model to split the sentence set of the given papers into different topic-biased parts, and then applies regression models to learn the importance scores of the sentences. At last an optimization framework is proposed to generate the related work section. Evaluation results show that our system can generate much better related work sections than the baseline methods.

In future work, we will make use of citation sentences to improve our system. Citation sentences are the sentences that contains an explicit reference to another paper and they usually highlight the most important aspects of the cited papers. So citation sentences are likely to contain important and rich information for generating related work sections.

Acknowledgments

The work was supported by National Natural Science Foundation of China (61170166, 61331011), Beijing Nova Program (2008B03) and National Hi-Tech Research and Development Program (863 Program) of China (2012AA011101). We also thank the anonymous reviewers for very helpful comments. The corresponding author of this paper, according to the meaning given to this role by Peking University, is Xiaojun Wan.

Reference

Nitin Agarwal, Kiran Gvr, Ravi Shankar Reddy, and Carolyn Penstein Rosé. 2011. Towards multi-document summarization of scientific articles: making interesting comparisons with SciSumm. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pp. 8-15. Association for Computational Linguistics.

Phyllis B. Baxendale. 1958. Machine-made index for technical literature: an experiment. *IBM Journal of Research and Development* 2, no. 4: 354-361.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 481-490. Association for Computational Linguistics.

Chih-Chung Chang, and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, no. 3: 27.

John M. Conroy, and Dianne P. O'leary. 2001. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 406-407. ACM.

Harold P. Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)* 16, no. 2: 264-285.

Günes Erkan, and Dragomir R. Radev. 2004. LexPageRank: Prestige in Multi-Document Text Summarization. In *EMNLP*, vol. 4, pp. 365-371.

Günes Erkan, and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.(JAIR)* 22, no. 1: 457-479.

Dimitrios Galanis, Gerasimos Lampouras, and Ion Androutsopoulos. 2012. Extractive Multi-Document Summarization with Integer Linear Programming and Support Vector Regression. In *COLING*, pp. 911-926.

Dimitrios Galanis, and Prodromos Malakasiotis. 2008. Aueb at tac 2008. In*Proceedings of the TAC 2008 Workshop*.

Dan Gillick, and Benoit Favre. 2009. A scalable global model for summarization. In*Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pp. 10-18. Association for Computational Linguistics.

Cong Duy Vu Hoang, and Min-Yen Kan. 2010. Towards automated related work summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 427-435. Association for Computational Linguistics.

Lei Huang, Yanxiang He, Furu Wei, and Wenjie Li. 2010. Modeling document summarization as multi-objective optimization. In *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, pp. 382-386. IEEE.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50-57. ACM.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In*Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74-81.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, no. 2: 159-165.

Inderjeet Mani, and Eric Bloedorn. 1999. Summarizing similarities and differences among related documents. *Information Retrieval* 1, no. 1-2: 35-67.

Ryan McDonald. 2007. *A study of global inference algorithms in multi-document summarization*. Springer Berlin Heidelberg.

Qiaozhu Mei, and ChengXiang Zhai. 2008. Generating Impact-Based Summaries for Scientific Literature. In *ACL*, vol. 8, pp. 816-824.

Rada Mihalcea, and Paul Tarau. 2005. A language independent algorithm for single and multiple document summarization.

Rada Mihalcea, and Hakan Ceylan. 2007. Explorations in Automatic Book Summarization. In *EMNLP-CoNLL*, pp. 380-389.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 584-592. Association for Computational Linguistics.

Preslav Nakov, Ariel Schwartz, and M. Hearst. 2004. Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*.

Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. 2008. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 542-550. ACM.

Vahed Qazvinian, and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 689-696. Association for Computational Linguistics.

You Ouyang, Sujian Li, and Wenjie Li. 2007. Developing learning strategies for topic-based summarization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 79-86. ACM.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek et al. 2004. MEAD-a platform for multidocument multilingual text summarization. Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004).

Ariel S. Schwartz, and Marti Hearst. 2006. Summarizing key concepts using citation sentences. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, pp. 134-135. Association for Computational Linguistics.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document Summarization Using Conditional Random Fields. In *IJCAI*, vol. 7, pp. 2862-2867.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics* 26, no. 3: 339-373.

Kristian Woodsend, and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 233-243. Association for Computational Linguistics.

Shasha Xie, Benoit Favre, Dilek Hakkani-Tür, and Yang Liu. 2009. Leveraging sentence weights in a concept-based optimization framework for extractive meeting summarization. In *INTERSPEECH*, pp. 1503-1506.

Hui Yang, and Jamie Callan. 2009. A metric-based framework for automatic taxonomy induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pp. 271-279. Association for Computational Linguistics.

Ozge Yeloglu, Evangelos Milios, and Nur Zincir-Heywood. 2011. Multi-document summarization of scientific corpora. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pp. 252-258. ACM.