

# CIST@CLSciSumm-17: Multiple Features Based Citation Linkage, Classification and Summarization

Lei Li, Yazhao Zhang, Liyuan Mao, Junqi Chi, Moye Chen, and Zuying Huang

Center for Intelligence Science and Technology (CIST), School of Computer  
Beijing University of Posts and Telecommunications (BUPT), Beijing, P.R.China  
{leili,yazhao,circlelyuan,cjq,moyec,zoehuang}@bupt.edu.cn

**Abstract.** This paper describes our methods and experiments applied for CLSciSumm-17. We try Convolutional Neural Network, word vectors and sentence similarities for citation linkage. For facet classification, we explore more useful features, rules, SVM and Fusion method. We use the linear combination of five classical features and DPPs based diversity sampling method to compute the structured summary. Test-data results show that we obtain the best performance for both facet classification and summarization.

**Keywords:** CNN, word vector, SVM, hLDA, DPP

## 1 Introduction

With the rapid development of Computational Linguistics (CL), rich, complex and continually expanding resource has flooded into the scientific literature of this domain. Literature surveys and review articles in CL do help readers obtain a gist of the state-of-the-art in research for a topic. However, literature survey writing is labor-intensive and one specific literature survey is not always available for every topic of interest. The CLSciSumm-17 has highlighted the challenges and relevance of the scientific summarization problem.

In this paper, we describe our strategies, methods and experiments applied for CLSciSumm-17 [1]. There are totally two tasks and 30 topics involving one reference paper (RP) and some citing papers (CPs) for the training dataset. The tasks are defined as follows: Given a set of CPs that all contain citations to a RP. Given a set of CPs that all contain citations to a RP for each topic, the tasks are defined as follows. **Task 1A** requires that for each citance (the set of citation sentences), we need to identify the spans of text (cited text spans, CTS) in the RP. We will try to use Convolutional Neural Network(CNN) with word vectors to match the citance and possible CTS. Besides, we also have existed methods using sentence similarity based on various traditional features and rules. We will search for a good hybrid methods which can take advantages of both CNN and the existed methods.

Based on the results of Task 1A, for each cited text span, we need to identify

what facet the CTS belongs to, from a predefined set of facets in **Task 1B**. We plan to explore more useful features and machine learning methods besides SVM.

Finally, we will generate a structured summary of the RP and all of the community discussion of the paper represented in the citances for **Task 2**. The length of the summary should not exceed 250 words. We will extract summaries with high quality, diversity and low redundancy. hLDA (hierarchical Latent Dirichlet Allocation) topic model is adopted for content modeling, which is able to organize topics into a tree-like structure. We combine hLDA knowledge with several other classical features using different weights and proportions to evaluate the quality. And the summary diversity is enhanced using Determinantal Point Processes (DPPs).

## 2 Related Work

Interest about information extraction and retrieval has increased recently, and there are some shared tasks in this domain like Online Forum Summarization (OnForumS) in MultiLing 2017 and Computational Linguistics Scientific Document Summarization Shared Task. Through these shared tasks, many methods have been found for content linking, Ziqiang Cao et al. [2] simplify it into a ranking problem which just needs to select the first item, then they adopt SVM Rank to handle it. Bruno et al. [3] transform each word into their synonym group synset using WordNet for next idf (inverse document frequency) calculation step. Kun Lu et al. [4] use WordNet to compute the concept similarity between sentences. Tadashi [5] combines TFIDF and a single layer Neural Network together for content linking.

In order to identify the linkage efficiently between a paper citation and its cited text spans in the RP, we need to catch the potential information of natural language sentences. More and more methods have been proposed for this purpose. CNN [6] has shown good performance in sentence classification, Word Embedding [7] is useful in digging semantic information, WordNet is used to calculate the similarity between words. All of above will be used in our experiments.

In the case of CL summary generation, we will not only consider the original paper, but also consider the information provided by citances. Li et al. [8] provide a method based on linear combination of multiple features. Amjad et al. [9] focused on the coherence and readability of generated citation summaries. Yet their objects for summarization are different from ours, because we are focusing on summarizing the reference texts considering the citation texts in this paper. It is a common view that summaries should consider redundancy as well as quality. But, unfortunately most previous methods divide the summary generation problem into two procedures in sequence. First, they use some machine learning methods [13] to select a subset with higher quality. Then, they control the redundancy of summaries [14]. In our method, we will consider diversity, redundancy and quality at the same time. And we will use a new method based on DPPs [12] to enhance the diversity of summaries.

## 3 Methods

### 3.1 Citation Linkage

For citation linkage, we need to identify the CTS in the RP that most accurately reflect the citation in the CP, we call this content linking. In fact, it focuses on finding the linkage between sentences, which is represented by similar meaning between sentences. Hence computing sentence similarity based on various features is our major work. For features, we use some traditional features like Jaccard similarity and idf to find syntactic information. Besides, word vector, WordNet and CNN are used for digging out deeper semantic information. Finally, we fuse the above features to obtain the result.

#### Feature Extraction

1) Three lexicons: a) we picked up the words with high frequency from reference text in the training corpus artificially, and expanded them through WordNet and word vectors as Lexicon 1 (high-frequency lexicon). b) we used LDA (Latent Dirichlet Allocation) model to train the reference paper and citing papers to obtain a lexicon of 30 latent topics for files in every topic independently as Lexicon 2 (LDA lexicon). c) we obtained the co-occurrence degree between words by the word frequency statistics of citation text and its reference text from the training corpus as Lexicon 3 (co-occurrence lexicon).

2) Two sentence similarities: One is idf similarity, we add up the idf values of the same words between two sentences. The other is Jaccard similarity, which uses the division between the intersection and the union of the words in two sentences as similarity.

3) Two context similarities: We calculate the context similarity for idf similarity and Jaccard similarity. The  $F_1$  performances of the above features for training corpus are shown in Table 5 of Appendix B.

4) Word vector: We trained every word as a vector with fixed dimensions using Word2Vec. Then add the word similarities together to represent sentence similarity [8]. The word vector of 400 dimensions with window size as 15 performs best as shown in Table 6 in Appendix B, so we will choose it for the following experiments.

5) WordNet: WordNet is a kind of English lexicon which contains nouns, verbs, adjectives, and adverbs. It can calculate the similarity between two words with same part-of-speech. We use 6 similarity methods: jcn, lin, lch, res, wup and path similarity. Though the similarity in WordNet can only support calculating word similarity, in order to compute the sentence similarity, we use the same algorithm in word vector section for WordNet similarity, changing the cosine similarity to word similarity in WordNet. The  $F_1$  performances can be seen in Table 7 of Appendix B.

6) CNN: CNN can find the deep semantic information in sentences and is widely used in natural language processing domain [11]. We use the word vector as the input of CNN and obtain the probability of content linking from its output.

We use the output probability to represent the similarity of input sentences. We have investigated the lengths of reference sentences and citation ones, almost all of them are less than 80. So we fix the length of sentence to 80 and the length of two sentences is 160. For the sentence whose length is less than 80, we add 0 vector to the relevant location. We use the word vector with 200 dimensions. As for the CNN structure, we have tried some kinds of configurations involving different convolution kernels and other parameters. Finally, through the experiments on the training corpus, we use one of them for testing corpus.

### Method for Linkage

1) For voting method, we tried different weights(the weights of different features) and proportions(the number of sentences we choose) to combine them through experiments, and then got four results (run 1, run 2, run 5, run 6) through a voting system. The text span with the highest-number of votes is chosen as the citation text corresponding sentences in the reference paper. Jaccard Focused Method (run 3) chose Jaccard similarity as the major feature, and added other features as supplementary. Jaccard Cascade Method (run 4) chose the sentences with top two Jaccard values as the basic answer, then combined other features to find the other two sentences with highest values as the improved answer. Finally, we choose 4 sentences as answer through experiments. Table. 1 shows the parameters for every run. W means weight and P means proportion.

**Table 1.** The parameters of run 1 to 6, JS means 10 fold of Jaccard Similarity

	run1		run2		run3		run4		run5		run6	
Feature	W	P	W	P	W	P	W	P	W	P	W	P
Idf similarity	1	8	1	7	0.7	15	1.5	16	1.5	15	0.8	15
Idf context similarity	-	-	0.5	4	0.5	15	1	18	1.5	20	0.5	20
Jaccard similarity	1	12	1	3	<i>JS</i>	7	-	-	2	10	1	8
Jaccard context similarity	1	12	0.5	8	0.7	15	1.5	15	2	15	0.7	15
Word vector	1	10	0.5	8	0.5	25	-	-	0.8	10	0.5	15
Lexicon 2	-	-	0.3	2	-	-	-	-	-	-	-	-
Lexicon 3	-	-	0.4	2	0.2	25	0.5	15	0.5	15	0.5	15
Lch	-	-	-	-	-	-	-	-	1	10	0.5	10
Path	-	-	-	-	-	-	-	-	1	10	0.5	10

2) For method with CNN: We use the answers in training data as positive samples, and choose the sentences out of answers randomly as negative samples, keeping the count of them in balance. For testing set, we combine every sentence in reference paper with one citation text as input, getting the output as feature value. While only using this feature, the performance is bad, so we choose the following method for CNN: use top 40 sentences according to CNN output, then combine Jaccard similarity and idf similarity together using Equation 1. All

parameters were obtained through experiments on training corpus.

$$similarity = 10 * Jaccard_{similarity} + 0.1 * idf_{similarity} \quad (1)$$

Finally, we choose the top 4 sentences with highest similarities as answers.

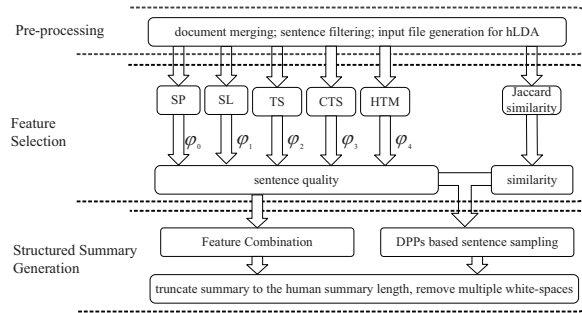
### 3.2 Facet Classification

Subtitle Rule obtains Facet Classification according to the subtitle of sentence pair. We use the high frequency word number of sentence pair to obtain the Facet Classification in High Frequency Word Rule. In Subtitle and High Frequency Word Combining Rule, we combine the two methods for Facet Classification. SVM Classifier uses features of sentence pair to obtain the classification. The features are Location of Paragraph, Document Position Ratio, Paragraph Position Ratio and Number of Citations or References. The ideas of Voting Method and Fusion Method are similar.

The details of above methods are mostly similar to [8] except for some details of High Frequency Word Rule, SVM Classifier and Fusion Method. The difference is that we update the High Frequency Word and retrain the SVM Classifier. In Fusion Method, we combine all the results to obtain a fusion result. We provide the details of features used in SVM Classifier in Appendix A.

### 3.3 Summarization

Figure. 1 shows our framework of Task 2.



**Fig. 1.** The framework for structured summary generation

**Pre-processing** The source documents provided by CLSciSumm-17 have relatively high quality, but there also exist some xml-coding errors. Besides, we need specific data format to train our hLDA feature.

1. Document Merging: merge the content of RP and the citations into a document. It is worth mentioning that we will not include the sentence in the abstract of RP unless it is selected by Task 1A. And all documents are converted to lowercase letters.
2. Sentence filtering: the corpus is generated from CL papers, thus there must be some equations, figures, tables and so on. However, these contents make small contribution to summary generation. First, we use WordNet to check whether a word is useful. Then we can filter those sentences whose proportion of useless words is greater than 0.5.
3. Input file generation for hLDA: For the remaining words, we build a dictionary for each document, which contains words and their corresponding frequencies, and the index starts from 1 to word list size. Finally, we generate an input file for hLDA modeling, in which each line represents a sentence presented by word index - frequency pair, such as:

$[number\ of\ words\ in\ sentence_i] [word - index\ A : frequency\ A] \dots$

**Feature Selection** According to our previous work [8], Sentence Length (SL), Sentence Position (SP) and CTS are useful features for summary generation of CL papers and we will reuse them in this paper. Besides, we upgrade the hierarchical Topic Model (HTM) and apply title similarity (TS) to our experiments.

1. *hierarchical Topic Model (HTM)*: hLDA constructs a document into a tree-like structure. Each word is assigned to a node, and each sentence is assigned to a path that goes through the root node to a leaf node. Thus, each node is regarded as a topic which is a probability distribution over words, and each path is always regarded as a theme. Based on this, we believe that both topics and themes involve valuable message for summary generation. We adapt the method in [10] to use the information of hLDA. Different from the method proposed by Taiwen [10], we also consider the theme distribution to measure the contribution of a sentence to the summary. We use Equation. 2 to calculate it.

$$q_i = \sum_{j=1}^m (\alpha_j T_j + Freq_j) + tp_i \quad (2)$$

where  $T_j$  represents the distribution score of  $word_j$  in  $i$ -th sentence calculated by hLDA,  $\alpha_j$  is the pre-defined weight of  $T_j$  according to our former experiments,  $Freq_j$  is the frequency of  $word_j$  in current hLDA node,  $tp_i$  is the theme distribution of  $i$ -th sentence.

2. *Title Similarity (TS)*: Title Similarity is the cosine similarity of each sentence and the document title. We use Equation (3) to calculate it.

$$q_i = \frac{tf_{s_i} \times tf_{s_{title}}}{|tf_{s_i}| |tf_{s_{title}}|} \quad (3)$$

where  $s_{title}$  and  $s_i$  represent the title and  $i$ -th sentence respectively.

**Structured Summary Generation** To control the redundancy of the generated summary, we use Jaccard similarity to measure the similarity between sentences firstly. Then we use DPPs to enhance the diversity.

Determinantal Point Processes (DPPs) are elegant probabilistic models of global, negative correlations and mostly used in quantum physics to study the reflected Brownian motions. In our method, we only consider discrete DPPs and follow the definition of Kulesza [12].

DPPs based subset sampling method considers diversity, quality and redundancy at the same time. And the subset sampled using DPPs are mostly diverse. Thanks for the contribution of [15], we can construct DPPs using a positive semi-defined matrix  $L$ .

Using this representation, the entries of kernel  $L$  can be written as

$$L_{ij} = q_i \phi_i^\top \phi_j q_j$$

where  $q_i \in R^+$  measures the *quality* of an element  $i$ , and  $\phi_i^\top \phi_j$  measures the *similarity* between element  $i$  and element  $j$ .

In our method, we use the combination of previous five features (SP, SL, TS, CTS, HTM) to measure the quality, and Jaccard similarity to measure the redundancy, shown as below.

$$q_i = \sum_{k=1}^5 \varphi_k q_{ki} \quad (4)$$

where  $\varphi_k \in \{0, 1\}$  is the combination proportion of corresponding features.  $q_{ki}$  represents the  $k$ -th quality feature of  $s_i$ . For example  $q_{11}$  represents the second feature (SL) of second sentence in the article.

Given the matrix  $L$ , we adapt the DPPs based sampling method to compute a sentence subset  $D'$  of the corresponding paper, shown in Table. 2.

Finally, in order to extract a high-quality structured summary, we make full use of the prior knowledge that structured summary contains four parts: Introduction, Methods, Results and Conclusion. So we use the Facet obtained in Task 1B to help extract the summary sentences from  $D'$ . We will extract two or three sentences for each part of the summary if exists. Furthermore, we remove redundant candidate sentences.

Besides, we also use  $q_i$  calculated using Equation. 4 without DPP to select top-N sentences to compute a summary. We call this method as Feature Combination method(FC). This is a comparison system to examine the effectiveness of DPPs based sampling method.

## 4 Experiments

Table. 3 shows the official results of 7 runs we submitted.

### 4.1 Citation Linkage and Facet Classification

From Run1 to Run6, we use the above mentioned methods respectively for Task 1A. In Rnn 7, we use the CNN method, while it uses all corpus for training, we

**Table 2.** DPPs sampling method for diverse summary extraction, where  $S$  is the similarity matrix,  $D$  is the document,  $q_i$  is the quality of  $i$ -th sentence calculated in Equation. 4

---

**Input:**  $q_i, S, D, \text{max\_len}$ .  
 $\rightarrow \text{quality\_vec} = [q_i \text{ for } i \text{ in } D]$   
 $\rightarrow \text{matrix\_l} = \text{quality\_vec} * S * \text{quality\_vec}^T$   
 $\rightarrow (\mathbf{v}_n, \lambda_n) = \text{eigen\_decompose}(\text{matrix\_l})$   
 $\rightarrow J = \emptyset$   
 $\rightarrow \text{for } n = 1, 2, \dots, N \text{ do}$   
 $\rightarrow \rightarrow J = J \cup \{n\}$  with prob.  $\frac{\lambda_n}{\lambda_n + 1}$   
 $\rightarrow V = \{\mathbf{v}_n\}_{n \in J}$   
 $\rightarrow Y = \emptyset$   
 $\rightarrow \text{while } |V| > 0 \text{ and } \text{len}(Y) < \text{max\_len} \text{ do}$   
 $\rightarrow \rightarrow \text{Select } i \text{ from } J \text{ with } Pr(i) = \frac{1}{|V|} \sum_{v \in V} (\mathbf{v}^\top \mathbf{e}_i)^2$   
 $\rightarrow \rightarrow Y = Y \cup D[i]$   
 $\rightarrow \rightarrow V = V_\perp$ , an orthonormal basis for the subspace of  
 $V$  orthogonal to  $\mathbf{e}_i$

**Output:** summary  $Y$

---

**Table 3.** Results on Test Data

Runs	F Score of Task 1A		F Score of Task 1B		ROUGE 2 of Task 2		
	macro	micro	macro	micro	Abstract	Human	Community
run1	0.113	<b>0.107</b>	0.373	<b>0.107</b>	0.341	0.173	0.187
run2	0.086	0.067	0.326	0.067	0.322	0.225	0.195
run3	0.097	0.086	0.365	0.086	0.327	<b>0.275</b>	<b>0.204</b>
run4	<b>0.117</b>	0.105	<b>0.408</b>	0.105	<b>0.351</b>	0.156	0.184
run5	0.109	0.100	0.392	0.100	0.318	0.153	0.192
run6	0.091	0.077	0.330	0.077	0.331	0.184	0.185
run7	0.100	0.084	0.351	0.084	0.240	0.170	0.163

have no result about it [8].

We evaluate our methods on the training dataset using the evaluation scripts offered in the CL-SciSumm Shared Task 2017 official website. We evaluate six methods for Facet Classification on four results of Task 1A, and selected the best method for each result of Task 1A. We find out that the best Facet Classification of Task 1A result is obtained by the voting method except for the result of Jaccard-Focused obtained by High Frequency Word Rule.

From the official results in Table 3, we can see that our methods for test data performs very well. Run4 has performed the best for Macro Average  $F_1$ .

## 4.2 Summarization

According to our experiments on the training dataset, we select seven different parameter settings to calculate the testing dataset and obtain seven runs for



the CLSciSumm-17 competition. We select the human written summary as the only gold summary and the Manual ROUGE values of our experiments on the training dataset are shown in Table. 4.

**Table 4.** The ROUGE-1 values for training dataset

id	type	$\varphi_0$	$\varphi_1$	$\varphi_2$	$\varphi_3$	$\varphi_4$	human	community
run1	FC	3	0	2	1	0	0.44349	0.46051
run2	FC	1	0	0	0	0	0.43771	0.42914
run3	FC	<b>1</b>	<b>0</b>	<b>1</b>	<b>3</b>	<b>0</b>	0.44304	<b>0.47556</b>
run4	DPPs	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0.44696</b>	0.39661
run5	DPPs	3	0	2	1	1	0.43848	0.40638
run6	DPPs	3	0	0	1	0	0.44598	0.40549
run7	DPPs	1	0	1	3	0	0.44276	0.40126

Table. 4 shows that, all FC methods perform better than DPPs on community summary. However, DPPs mostly perform better than FC on human summary. The golden standard summary provided by CLSciSumm-17 includes the summary from the cited text span. CTS feature chooses sentence from this point, so run3 obtains the best precision as our anticipation. DPPs attempt to sample sentences from a global perspective. They consider diversity, quality and redundancy at the same time. That is to say, this inspiration happens to coincide with the way our human beings address the same problem.

During our experiments, we also find some interesting results. For example, DPPs perform better than FC in the case of same parameter settings. Table. 3 shows that run4 performs best in the results of DPPs based methods. Thus we believe that the clusters modeled by unsupervised hLDA can implicate some latent topic messages, although they are not directly matched to the pre-defined five facets, which leads to the poor performance in this pre-defined structured summarization task. But we think that it possibly will work better for other open domains without pre-defined structures especially in multilingual case and will achieve better results combining with DPPs.

In fact, among all the systems participating Task 2, our run3 has also performs the best on ROUGE 2 vs Human abstract and Community abstract, and our run4 has also performs the best on ROUGE 2 vs Abstract. All these results have verified the effectiveness of our methods again.

## 5 Conclusion and Future Work

The results of official test data have proved that the performance of methods we proposed is excellent. In Task 1A we tried to find a calculating similarity method to represent the true relationship between two sentences. For Task 1B we used fusion method to obtain a fusion facet classification. Finally, we considered the quality features, redundancy feature and diversity of RP and cited text

spans in Task 2. And we will also try to find some better ways to use more semantic features for citation linkage. Furthermore, we will continue finding a better method to choose the least sentences covering the most information.

## References

1. Kokil Jaidka, Muthu Kumar Chandrasekaran, Devanshu Jain, and Min-Yen Kan. "Overview of the CL-SciSumm 2017 Shared Task", In Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017), Tokyo, Japan, CEUR. (2017)
2. Cao, Z., Li, W., Wu, D.: PolyU at CL-SciSumm 2016. In: Proc. of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2016). pp. 132138. Newark, NJ, USA (June 2016)
3. Malenfant, B., Lapalme, G.: RALI System Description for CL-SciSumm 2016 Shared Task. In: Proc. of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2016). pp. 146155. Newark, NJ, USA (June 2016)
4. Lu, K., Mao, J., Li, G., Xu, J.: Recognizing reference spans and classifying their discourse facets. In: Proc. of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2016). pp. 139145. Newark, NJ, USA (June 2016)
5. Nomoto, T.: NEAL: A neurally enhanced approach to linking citation and reference. In: Proc. of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2016). pp. 168174. Newark, NJ, USA (June 2016)
6. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Computer Science. (2013).
8. Li L, Mao L, Zhang Y, et al. Computational linguistics literature and citations oriented citation linkage, classification and summarization[J]. International Journal on Digital Libraries:1-18.
9. Abu-Jbara, A., Radev, D.: Coherent citation-based summarization of scientific papers. In: Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 500-509. Portland, Oregon (2010).
10. Huang, T., Li, L., Zhang, Y.: Multilingual multi-document summarization based on multiple feature combination (2016)
11. Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12: 2493-2537 (2011)
12. Alex, K., Ben, T.: Determinantal point processes for machine learning. arXiv preprint arXiv:1207.6083 (2012)
13. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artificial Intelligence Review 47(1), 1-66 (2017)
14. Kam-Fai, W., Mingli, W., Wenjie, L.: Extractive summarization using supervised and semi-supervised learning. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. pp. 985-992. (2008)
15. Borodin, A.: Determinantal point processes (2009)

## Appendix A

The meaning of features in Task 1B:

- 1) Location of Paragraph: the order number of the paragraph in which the sentence is located.
- 2) Document Position Ratio: the ratio of sentence Sid to the total sentence number of the corresponding document.
- 3) Paragraph Position Ratio: the ratio of sentence Ssid to the total sentence number of the corresponding paragraph.
- 4) Number of Citations or References: the number of Citation Offset or Reference Offset.

The details of methods:

### 1 Rule-based method

1.1 Subtitle Rule: First of all, we examine whether the subtitle of reference sentences and citance contains the following facet words: Hypothesis, Implication, Aim, Results and Method. If the subtitle contains any one of these words, it will be directly classified as the corresponding facet. If it contains more than one of these words, it will be classified into all the facets. Else if it contains none of them, we just classify it as the facet of Method.

1.2 High Frequency Word Rule: According to the High Frequency Word Rule, we firstly count the High Frequency Word of five facets from the Training Set and the Development Set. In order to improve the coverage of sentences, we expanded the High Frequency Word to get some similar words of each facet. We set an appropriate threshold for each facet. If the number of the High Frequency Word of any facet in the sentence is more than the corresponding facet threshold, then we just use the facet whose coverage is the highest as the final class. if some facets' overage are same, then we just classify according to the sequence of Hypothesis, Implication, Aim, Results and Method. If all facets have not reached the threshold of each facet, we classify it as the Method.

1.3 Combine Subtitle and High Frequency Word Rule: We firstly use the Subtitle Rule to classify the testing set. If the results are not in the five facets of Hypothesis, Implication, Aim, Results and Method, then we use the High Frequency Word Rule to get the final facet.

### 2 SVM Classifier

We extract four features of sentence for each class. Then features from citance sentence and reference sentence form an 8-dimension vector of a pair of reference sentence and citation sentence. We train SVM to get five classifiers. For solving the problem of unbalanced training data, we set different weights for different classes. If we cannot get any class of the five facets, then we classify it as Method class.

### 3 Voting Method

We combine the results from Subtitle Rule, High Frequency Word and SVM classifier to generate the voting results with most votes.

### 4 Fusion Method

We run the above methods for each run result we obtained in Task 1A and

choose a best one as the final result. Then we also tried a fusion method to combine all the run results of the above methods obtained in Task 1A. We counted the number of Method, Results, Aim, Hypothesis and Implication, and set an appropriate threshold for each facet class to get a final result of facet class.

## Appendix B

**Table 5.** The performance of single syntactic information feature

Feature	N=3	N=4	N=5
high-frequency (Lexicon 1)	0.00592	0.00728	0.01028
LDA (Lexicon 2)	0.01999	0.02549	0.02674
co-occurrence (Lexicon 3)	0.22724	0.21663	0.20257
Idf similarity	0.08808	0.08920	0.09152
Idf-context similarity	0.06736	0.06735	0.06478
Jaccard similarity	0.09400	0.09102	0.08740
Jaccard-context similarity	0.05996	0.05947	0.06015

**Table 6.** The performance of different word vectors

Feature	N=3	N=4	N=5	Feature	N=3	N=4	N=5
15, 400	0.07698	0.07828	0.07661	10, 200	0.069578	0.07160	0.07147
15, 300	0.07550	0.07585	0.07455	8, 400	0.074760	0.07646	0.07404
15, 200	0.074012	0.07100	0.06941	8, 300	0.074759	0.07524	0.07609
10, 400	0.076980	0.07828	0.07609	8, 200	0.071058	0.07039	0.07352
10, 300	0.076240	0.07403	0.07352	-	-	-	-

**Table 7.** The performance of different similarities in WordNet

Feature	N=3	N=4	N=5	Feature	N=3	N=4	N=5
Jcn	0.02147	0.03095	0.03290	Res	0.02147	0.03095	0.03290
Lin	0.02147	0.03095	0.03290	Path	0.06292	0.06675	0.06684
Lch	0.04885	0.05097	0.04884	Wup	0.04145	0.04430	0.04216