NJUST @ CLSciSumm-17

Shutian Ma[1], Jin Xu[1], Jie Wang[1], Chengzhi Zhang[1,2,*]

[1] Department of Information Management, Nanjing University of Science and Technology,
Nanjing, China, 210094
[2] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), Fuzhou, China, 350108
mashutian0608@hotmail.com, 1292050078@qq.com, 1342234559@qq.com,
zhangcz@njust.edu.cn

Abstract. This paper introduces NJUST system which is applied in the CL-SciSumm 2017 Shared Task at the BIRNDL 2017 Workshop. The training corpus contains 10 articles of training set, 10 articles of development set and 10 articles of test set from CL-SciSumm 2016. Articles were created by randomly sampling documents from the ACL Anthology corpus and selecting their citing papers. In Task 1A, we utilize different measurements to compute sentence similarities. Four classifiers are trained using different features and final results are obtained by voting system. In Task 1B, rule-based methods are mainly used according to high frequency words. As to Task 2, we generate a summary within 250 word based on the identified sentences in the reference paper from its cited text spans using maximal marginal relevance.

1 Introduction

Scientific papers are usually measured by their citances in citing papers which reveal the extent to which a reference paper has been used by other researchers. So far, most investigation has been focused on citation analysis from using simple index of citation counts [1, 2, 3] to complex natural language processing of citation contents [4]. However, using citances can't provide context from the reference paper, for example, the type of information cited or where it is in the referenced paper. To understand different perspectives of a reference paper, it's important to generate summary from all the cited text spans in the reference paper from citations [5, 6, 7, 8]. The CL-SciSumm 2017[2] has been designed to do automated summarization of scientific contributions for the computational linguistics research domain, which can help readers to gain a gist of the state-of-the-art in research for a topic.

CL-SciSumm 2017 has been divided into two tasks. Firstly, we should identify text spans in reference paper which most accurately reflect citance, facets of paper are also needed to be distinguished. Second task is to generate a summary of reference paper from the identified cited text spans. In this paper, we describe our methods applied for

[*] Corresponding Author

[2] Available at: http://wing.comp.nus.edu.sg/~cl-scisumm2017/

CL-SciSumm 2017. As to Task 1A, we trained four classifiers and integrate all the results by voting system. In Task 1B, rule-based methods are mainly used on identified text span to determine which facet it belongs to. As to Task 2, we generate a summary using maximal marginal relevance.

2 Related Work

This year's CL-SciSumm 2017 takes place at the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017)[3] and is a follow-up on the shared task of CL-SciSumm 2016[4] [9]. Originally, the CL Summarization Pilot Task was conducted as a part of the Biomed-Summ Track at the Text Analysis Conference 2014 (TAC 2014)[5] [10]. There have been many investigations on task problem previously [11, 12, 13, 14, 15, 16].

When doing Task 1A, most teams identified the linkage between a paper citation in citing paper and the corresponding cited text spans in reference paper by computing sentences similarities. CIST system applied two kinds of features, one is from lexicons, and another is from sentence similarities [11]. Aggarwal and Sharma made use of sub-sequences (of words) overlap [13]. Bi-grams were identified between generated bag-of-words to find matching statement in their study. PolyU [12] utilized *TF-IDF* cosine similarity, position of sentence chunk and some lexical rules. SVM and its modification model were chosen as the classifier for many teams [11, 12, 15]. New models have also been proposed by combining new algorithms. Klampfl, Rexha and Kern proposed TextSentenceRank for extracting candidate text spans which is inspired by graph based ranking algorithms [16]. Nomoto introduced a composite model consisting of *TF-IDF* and Neural Network [14].

As for Task 1B, since the instances for the Implication and Hypothesis facets are very limited, some teams only trained classification model on data of the other three facets [12]. Machine learning models such as, decision tree [12], random forest classifier [16], and SVM [11] were applied to conduct classification. Lexical rules are mainly used on section headers or citance content [12, 13, 16]. Researchers will try to build word lists for each facet which are similar words within each list. And then, they will examine whether the subtitles of reference sentences or cited sentences contains the following facet words or not for identification.

Few teams took part in Task 2 of generating summary. CIST system calculated sentence scores of five features: hLDA-level distribution feature, sentence-length feature, sentence-position feature, cited text span and RST-feature. They also use discourse facet to extract best-N sentences from all the sentences or from each cluster [11]. PolyU [12] converted Task 2 into the query-focused multi-document summarization problem. They used improved manifold ranking by modifying the prior score distribution to inspect the importance of citances.

[3] Available at: http://wing.comp.nus.edu.sg/~birndl-sigir2017/

[4] Available at: http://wing.comp.nus.edu.sg/cl-scisumm2016/

[5] Available at: http://www.nist.gov/tac/2014

3 Methodology

3.1 Task Description

There are two tasks in CL-SciSumm 2017 and framework is shown in Figure 1. The training dataset contains 30 topics of documents. A topic is consisted of a Reference Paper (RP) and Citing Papers (CPs) that all contain citations to the RP. In each CP, the text spans (citances) have been identified that pertain to a particular citation to the RP. In Task 1A, for each citance, we need to identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. In Task 1B, for each cited text span, we need to identify what facet of the paper it belongs to, from five predefined facets, which are Aim, Method, Results, Implication and Hypothesis. In Task 2, we need to generate a structured summary of the RP from the cited text spans of the RP.

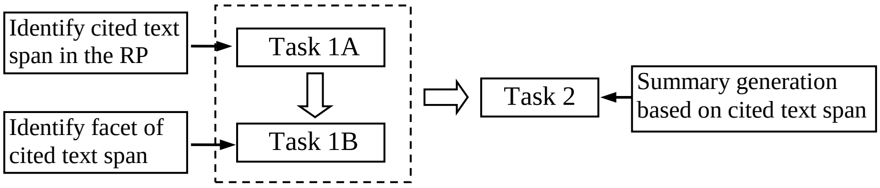

Identify cited text span in the RP
Task 1A
Identify facet of cited text span
Task 1B
Task 2
Summary generation based on cited text span

Fig. 1. Framework of Task 1A, Task 1B and Task 2

When doing evaluation, Task 1 will be scored by overlap of text spans measured by number of sentences in the system output and the gold standard created by human annotators. Task 2 will be scored using the ROUGE family of metrics.

3.2 Task 1A

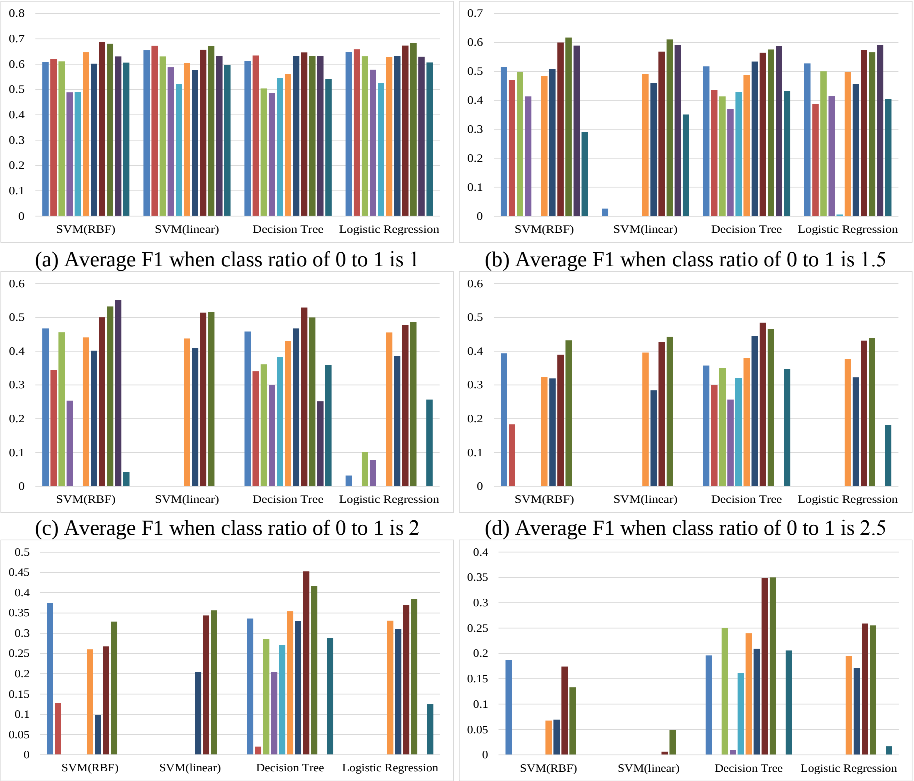
In this task, we are asked to identify the reference sentences referred to by a given citance. We approach this problem from the perspective of finding the sentence in RP which is more similar with citance and treat it as a classification task. In order to get better performance, we applied different classifiers and combined their results by voting system. In order to train the models, three kinds of features are obtained. Short descriptions of features are shown in Table 1.

Table 1. Three Kinds of Features Utilized in Task 1A

Feature Type	Feature	Feature Definition
Similarity-based features	LDA similarity	Cosine value between two sentence vectors trained by LDA
	Jaccard similarity	Division between the intersection and the union of the words in two sentences
	IDF similarity	Add up *IDF* values of the same words between two sentences
	TF-IDF similarity	Cosine value between two sentence vectors represented by *TF-IDF*
	Doc2Vec similarity	Cosine value between two sentence vectors trained by Doc2Vec

Rule-based features	Bigram	Bi-gram matching value, if there is bi-gram matched, the value is 1; otherwise, value is 0.
Position-based features	Sid	Sentence position in the full text
	Ssid	Sentence position in the corresponding section
	Sentence Position	The sentence position, divided by the number of sentences
	Section Position	The position of the corresponding section of the sentence chunk, divided by the number of sections
	Inner Position	The sentence position in the section, divided by the number of sentences in the section

Based on the annotation files, we give labels to the matched sentence pairs with 1 and unmatched sentence pairs with 0. When training classifiers, we firstly tried six different models, including SVM (kernel=linear), SVM (kernel=rbf), SVM (kernel=sigmoid), decision tree, logistics regression and nearest neighbor. Different features are investigated on all datasets of CL-SciSumm 2017. According to the 10 fold cross validation results, we remove SVM (kernel=sigmoid) and nearest neighbor, and choose different features for the remaining classifiers. The average F_1 values of all features for Task 1A on training dataset are shown in Figure 2. In order to find good features, we trained the classifiers for 8 runs with the different class ratios of 0 and 1 labels. From Figure 2 (a) to Figure 2 (h), the class ratio of 0 to 1 is 1, 1.5, 2, 2.5, 3, 5, 7.5, and 10.



(a) Average F1 when class ratio of 0 to 1 is 1

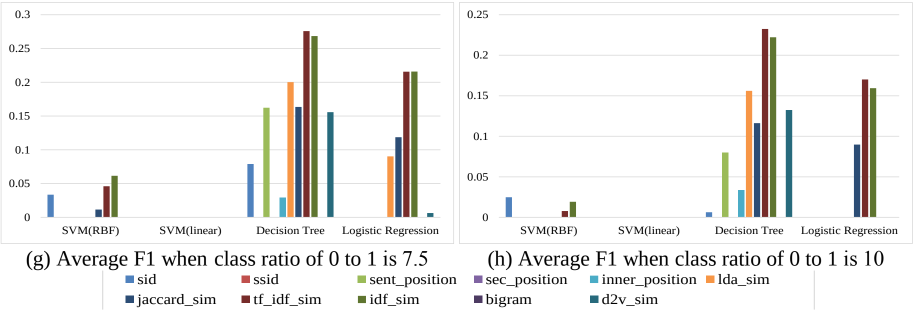
(b) Average F1 when class ratio of 0 to 1 is 1.5

(c) Average F1 when class ratio of 0 to 1 is 2

(d) Average F1 when class ratio of 0 to 1 is 2.5

(e) Average F1 when class ratio of 0 to 1 is 3

(f) Average F1 when class ratio of 0 to 1 is 5



(g) Average F1 when class ratio of 0 to 1 is 7.5

(h) Average F1 when class ratio of 0 to 1 is 10

Fig. 2. Average F1 of All Features for Task 1A with Different Proportion of 0/1 Sample Size

Based on these results, we can find that similarity-based features show better performance than the others. So we keep all similarity-based features, rule-based feature and choose some of the position-based features as the final features. Moreover, we set different weight to each classifier while all the results are integrated by voting system. Parameter settings are shown in Table 3.

Table 2. Parameter Setting of Different Classifiers

Classifier	Training features	Voting weight
SVM (kernel=linear)	LDA similarity, Jaccard similarity, *TF-IDF* similarity, *IDF* similarity, Doc2Vec similarity, Bigram, Ssid	0.25
SVM (kernel=rbf)	LDA similarity, Jaccard similarity, TF-IDF similarity, IDF similarity, Doc2Vec similarity, Bigram, sentence position, section position, inner position	0.4
Decision Tree	TF-IDF similarity, IDF similarity, Doc2Vec similarity, Bigram, Ssid, sentence position	0.15
Logistics Regression	TF-IDF similarity, IDF similarity, Doc2Vec similarity, Bigram, Ssid, sentence position	0.2

Due to the big quantitative gap between 1 and 0 labels, we trained the classifiers for 5 runs with the different proportion of 1 and 0 labels and set penalty factor as well. Furthermore, we also set different thresholds to the voting system. Detailed information of 1 and 0 label proportion and voting system thresholds in 5 runs is shown in table 3. Finally, according to the requirements of Task 1A, we did tuning on obtained results. For each citance, if the identified text spans contain more than 5 sentences, then we will list sentences in the order of Jaccard similarity from big to small, and pick the top 5 sentences to be the final results. If we can't identify any text span, then we will list sentences in the order of Jaccard similarity from big to small, and pick the top 1 sentence to be the final result.

Table 3. Detailed Information of Running Settings

Running Settings	0/1 sample size	Penalty Factor	Thresholds
Run1	5.5	5.5	0.8
Run2	4.5	4.5	0.8
Run3	6.5	6.5	0.8
Run4	5.5	5.5	0.7
Run5	5.5	5.5	0.6

3.3 Task 1B

In this task, for each cited text span, we need to identify what facet of the paper it belongs to. We construct three dictionaries of five facets Manual Dictionary, POS Dictionary-I and POS Dictionary-II. The first one is made manually and another two is made according to part-of-speech tagging results. Facet identification strategy of Task 1B is shown in Figure 3.

Referring to manual dictionary, we looked through each identified text span of five facets from all the annotation files in datasets. Then we build the dictionaries by judging every word within the sentence context manually. Two graduate students took part in this task.

Referring to POS dictionary, we firstly made part-of-speech tagging by Stanford POS Tagger[6] on the section title and sentence content in all the labeled annotation files. Then we keep the words which are adjectives and verbs and make all words as the automatic dictionary of section title and sentence content separately. We then list all words by frequency order according to five facets. After removing the words whose frequency is less than 2, the left words are the automatic dictionary of section title and sentence content separately. This is the POS dictionary-I. Since there are more words that related to method citation. We built POS dictionary-II by removing the method dictionary of section title and sentence content.

Based on the five different dictionaries of five facets, if the section title or sentence content contains any one of these words in the corresponding built dictionaries, it will be directly classified as the corresponding facet. Since the manual dictionary will be more accurate than POS dictionary. When using manual dictionary, identified facets will be all kept which means one sentence can have more than one facet. When using POS dictionary, the order of judging facet is hypothesis, aim, implication, method and result and later identified facet will override the former one. Finally, each sentence will have five identified facets, if five facets contain more than three of one facet, then we classify it as this facet. Else if it contains more than three different facets, we just classify it as the facet of Method.

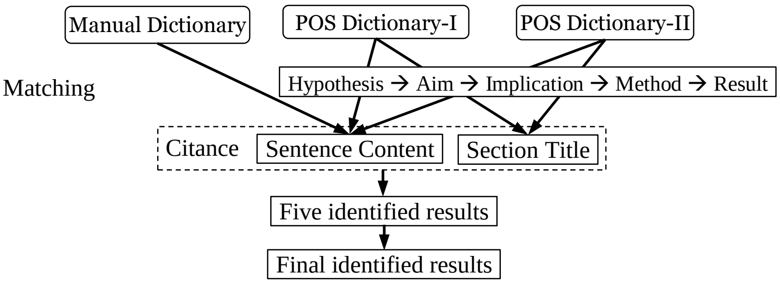


Fig. 3. Facet Identification Strategy of Task 1B

[6] Available at: https://nlp.stanford.edu/software/tagger.html

3.4 Task 2

Summary generation is divided into two main steps. First is to group sentences into different clusters by bisecting K-means [17]. Second is using maximal marginal relevance (MMR) [18] to extract sentence from each cluster and combine them into a summary.

Firstly, we use vector space model to represent documents and then non-negative matrix factorization is conducted to reduce the document dimension into 50 dimensions. Then we apply the bisecting K-means which is based on K-means. Bisecting K-means can be divided into four steps: 1.Pick a cluster to split; 2.Find 2 sub-clusters using the basic K-means algorithm; 3. Repeat step 2, the bisecting step, for a fixed number of times and take the split that produces the clustering with the highest overall similarity. (For each cluster, its similarity is the average pairwise document similarity, and we seek to minimize that sum over all clusters.); 4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached. After obtaining the clusters, we list all the clusters in the order of cluster size from big value to small value. And then, all the sentences within each cluster are listed in the order of MMR from big value to small value. The basic idea of MMR is straightforward [19]: if we have a set of items D and we want to recommend a subset $S_k \subset D$ ($where\ |S_k| = k\ and\ k \ll |D|$) relevant to a given query q. MMR proposes to build S_k by selecting s_j^* given $S_{j-1} = \{s_1^*, \cdots, s_{j-1}^*\}$ ($where\ S_j = S_{j-1} \cup \{s_j^*\}$) according to the following criteria:

$$s_j^* = \arg \max_{s_j \in D \setminus S_{j-1}} \left[\lambda \left(Sim_1(s_j, q) \right) - (1 - \lambda) \max_{s_i \in S_{j-1}} Sim_2(s_j, s_i) \right] \tag{1}$$

Where $Sim_1(\cdot,\cdot)$ measures the relevance between an item and a query, $Sim_2(\cdot,\cdot)$ measures the similarity between two items, and the manually tuned $\lambda \in [0,1]$ trades off relevance and similarity. In the case of s_1^*, the second term disappears.

Finally, for each time, we choose first two sentence from each cluster to build the summary before the length of summary exceeds 250 words.

4 Experiments

4.1 Task 1A

When doing corpora preprocessing, we remove the stop words and stem words to base forms by Porter Stemmer algorithm[7]. Then, we applied D2V model in Genism[8] and python package[9] of LDA model to represent documents. All the classifiers were done via Scikit-learn[10] python package. The source code of our system will be successively open on the Github website: *https://github.com/KingChristenson/NJUST-CL*.

For classification experiments, we split training dataset into two separate datasets: 10 articles of training set and 10 articles of development set from CL-SciSumm 2016

[7] Available at: http://tartarus.org/~martin/PorterStemmer/

[8] Available at: http://radimrehurek.com/gensim/index.html

[9] Available at: https://pypi.python.org/pypi/lda

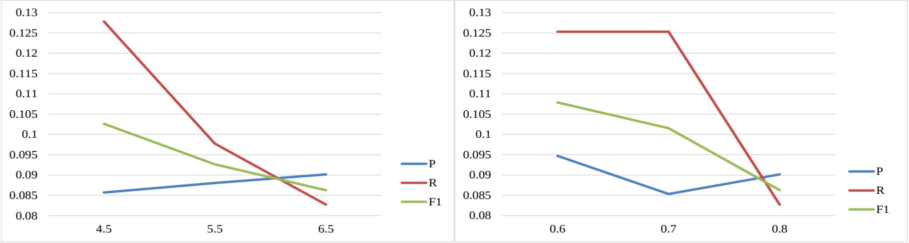
[10] Available at: http://scikit-learn.org/stable/index.html

are chosen as train dataset. 10 articles of test set from CL-SciSumm 2016 are chosen as test dataset. There are five runs that we submitted. Precision, Recall and F1 values which we got from the test dataset are shown in Table 4.

Table 4. Task 1A Results of Training Dataset

Running Settings	P	R	F_1
Run1	0.08804	0.09774	0.09264
Run2	0.08571	0.12782	0.10262
Run3	0.09016	0.08271	0.08627
Run4	0.08532	0.12531	0.10152
Run5	0.09470	0.12531	0.10787

We also draw Figure 4 (a) and Figure 4 (b) to see the trend of different evaluation results when increasing the class ratio of 0 to 1 and thresholds for voting system.



Note: *Blue line denotes precision, red line denotes recall and green line denotes F1 value.*

(a) Precision, Recall and F_1 when class ratio of 0 to 1 is Increasing

(b) Precision, Recall and F_1 when Threshold is Increasing

Fig. 4. Evaluation when Increasing Class Ratio of 0 to 1 and Threshold for Voting System

From Figure 4 (a), we can find that with the increasing of 0/1 sample size, although the precision value is increasing slowly, according to F1 value, the performance of Task 1A is getting worse. The same situation happened when we increasing the threshold. So it's important to choose the proper parameters in such classification tasks, such as the 0/1 sample size and threshold for voting system.

4.2 Task 1B

We tried all results from Task 1A, and then got the best performance by voting system for 5 runs. Table 5 shows our Task 1B results of the train data according to different facets.

Table 5. Task 1B Results of Training Dataset

Evaluation Facet	Precision	Recall	F_1
Aim Citation	0.16162	0.44444	0.23704
Implication Citation	0.50000	0.23256	0.31746
Hypothesis Citation	0.50000	0.50000	0.50000
Method Citation	0.74026	0.91566	0.81867
Result Citation	0.39286	0.48889	0.43564

From Table 6, we can find that identification of method citation performs best since it's also the most common facet shown in all citations. Citation facet of result, aim and implication shows bad performance. The poor quality of built dictionary might lead to this results. More features should be considered when doing this task, such as the sentence position or section title position.

5 Conclusion and Future Work

This document demonstrates our participant system NJUST on CL-SciSumm 2017. Our system has tried to add some semantic information like doc vector and topic distributions in LDA to improve the citance linkage and summarization performance. When choosing features, we find that TF-IDF similarity and IDF similarity do better than the similarities based Doc2Vec and LDA. In order to improve classification performance, several classifiers are trained with different features. The final results are obtained by voting system. When doing Task 2, we use maximal marginal relevance to rank sentences for summary generation. According to the evaluation [20], we did the best performance in Task 1A and also good in Task 1B, while strategy for Task 2 didn't work well and more work can be done in all the tasks.

In the future work, we need to find better ways to measure sentence similarities and use some machine learning models to do Task 1B. As to summarization, we will try to combine the sentence with its identified facet information for organizing the sentence order. Furthermore, more features can be added to calculate the sentence score for ranking, such as sentence length, sentence position, etc.

. Acknowledgements

This work is supported by Major Projects of National Social Science Fund (No. 16ZAD224), Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF201704) and Qing Lan Project.

Reference

1. Garfield E, Merton R K. Citation indexing: Its theory and application in science, technology, and humanities [M]. New York: Wiley, 1979.
2. Meho L I, Yang K. Impact of data sources on citation counts and rankings of LIS faculty: Web of Science versus Scopus and Google Scholar [J]. Journal of the Association for Information Science and Technology, 2007, 58(13): 2105-2125.
3. Bornmann L, Daniel H D. What do citation counts measure? A review of studies on citing behavior [J]. Journal of documentation, 2008, 64(1): 45-80.
4. Zhang G, Ding Y, Milojević S. Citation content analysis (cca): A framework for syntactic and semantic analysis of citation content [J]. Journal of the Association for Information Science and Technology, 2013, 64(7): 1490-1503.
5. Jaidka K, Khoo C S G, Na J C, et al. Deconstructing Human Literature Reviews-A Framework for Multi-Document Summarization [C]//ENLG. 2013: 125-135.

6. Nenkova A, McKeown K. Automatic summarization [J]. Foundations and Trends® in Information Retrieval, 2011, 5(2–3): 103-233.
7. Teufel S, Moens M. Summarizing scientific articles: experiments with relevance and rhetorical status [J]. Computational linguistics, 2002, 28(4): 409-445.
8. Jones K S. Automatic summarising: The state of the art [J]. Information Processing & Management, 2007, 43(6): 1449-1481.
9. Jaidka K, Chandrasekaran M K, Rustagi S, et al. Overview of the CL-SciSumm 2016 Shared Task [C]//BIRNDL@ JCDL. 2016: 93-102.
10. Jaidka K, Chandrasekaran M K, Elizalde B F, et al. The computational linguistics summarization pilot task [C]//Proceedings of Text Ananlysis Conference, Gaithersburg, USA. 2014.
11. Li L, Mao L, Zhang Y, et al. CIST System for CL-SciSumm 2016 Shared Task [C]//BIRNDL@ JCDL. 2016: 156-167.
12. Cao Z, Li W, Wu D. PolyU at CL-SciSumm 2016 [C]//BIRNDL@ JCDL. 2016: 132-138.
13. Aggarwal P, Sharma R. Lexical and Syntactic cues to identify Reference Scope of Citance [C]//BIRNDL@ JCDL. 2016: 103-112.
14. Nomoto T. NEAL: A Neurally Enhanced Approach to Linking Citation and Reference [C]//BIRNDL@ JCDL. 2016: 168-174.
15. Moraes L, Baki S, Verma R M, et al. University of Houston at CL-SciSumm 2016: SVMs with tree kernels and Sentence Similarity [C]//BIRNDL@ JCDL. 2016: 113-121.
16. Klampfl S, Rexha A, Kern R. Identifying Referenced Text in Scientific Publications by Summarisation and Classification Techniques [C]//BIRNDL@ JCDL. 2016: 122-131.
17. Steinbach M, Karypis G, Kumar V. A comparison of document clustering techniques[C]//KDD workshop on text mining. 2000, 400(1): 525-526.
18. Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries[C]//Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998: 335-336.
19. Guo S, Sanner S. Probabilistic latent maximal marginal relevance[C]//Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2010: 833-834.
20. Kokil Jaidka, Muthu Kumar Chandrasekaran, Devanshu Jain, and Min-Yen Kan (2017). 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.