

CIST@CLSciSumm-19: Automatic Scientific Paper Summarization with Citances and Facets

Lei Li, Yingqi Zhu, Yang Xie, Zuying Huang, Wei Liu,
Xingyuan Li, and Yinan Liu

Beijing University of Posts and Telecommunications (BUPT)
No.10 Xitucheng Road, Haidian District, Beijing, P.R.China
{leili,zhuyq,zoehuang,thinkwee,lynbngt}@bupt.edu.cn
xieyangsp@163.com kuukisann@gmail.com

Abstract. Starting from its former version, CIST@CLSciSumm-18, our CIST@CLSciSumm-19 system is going to participate in the shared Task 1A (citation linkage), Task 1B (facet classification) and Task 2 (summarization) in CLSciSumm-19@SIGIR2019. We mainly try to improve its methods for all the shared tasks. We build a new feature of Word2vec.H for the CNN model to calculate sentence similarity for citation linkage. We plan to adopt CNN and RNN variants for facet classification. And in order to improve the performance of summarization, we develop more semantic representations for sentences based on neural network language models to construct new kernel matrix used in Determinantal Point Processes (DPPs).

Keywords: Citation Linkage · Facet Classification · Summarization · Word2vec.H · Neural Network Language Model · DPPs · Determinantal Point Processes.

1 Introduction

As the scientific paper, computational linguistics has many characteristics such as professional knowledge, rigorous writing and strong logic. Reading such articles is very meaningful, but manual reading takes a lot of time, so we need to study how to extract good article summaries to reduce the workload of readers. The main work of CLSciSumm-19 [1] is to explore automatic summary methods based on the characteristics of the papers in the field of computational linguistics, and to provide a comprehensive and readable summary for the thesis.

We tried to solve the three tasks contained in CLSciSumm-19: Task 1A, Task 1B and Task2. The data set we use is the paper in the field of computational linguistics provided by the organizer. There are some topics in the dataset. A topic consisting of a Reference Paper (RP) and Citing Papers (CPs) that all contain citations to the RP. In each CP, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP. Task 1A: For each citance, identify the spans of text (cited text spans, CTS) in the RP that most accurately reflect the citance. These are of the granularity of a sentence

fragment, a full sentence, or several consecutive sentences (no more than 5). Task 1B: For each cited text span, identify what facet of the paper it belongs to, from a predefined set of facets. Task 2 (optional bonus task): Finally, generate a structured summary of the RP from the cited text spans of the RP. The length of the summary should not exceed 250 words.

In this paper, based on previous work, we add the Word2vec-H feature to the Task 1A method and used CNN to get the result of the content linking. For Task 1B, we use the improved CNN and RNN structures for classification. For Task 2, we develop more semantic representations for sentences based on neural network language models to construct new kernel matrix used in Determinantal Point Processes (DPPs).

2 Related work

Task 1A acts as a content linkage task, and the common method is to calculate similarity, which includes not only the Cosine similarity, the Jaccard similarity, and so on, but also some semantic similarity calculation methods, such as BM25 and VSM [2]. In addition, the various characteristics of the words are also very important, such as the position of the word, part of speech and frequency, etc. The characteristics of the words in the two sentences are added to the similarity calculation for the sentence-pair, and the similarity of the two sentences can be judged at the word level [3]. With the continuous expansion of the corpus and the increasing number of features, the machine learning method has begun to emerge for the task. Firstly, the researchers try the basic classifiers, such as SVM using a radial basis function kernel, Decision Tree and Logistic Regression to identify the reference span [4]. Various classifiers can learn different text features, integrating them together can reveal more text features. So the researchers use ensemble models, such as the Random Forest [3]. Besides, in order to more deeply explore the meaning of the sentence, deep neural networks are also applied, such as CNN [5] [6] [9], Siamese Deep learning Networks [8]. For Task1B, the rule-based method [8] [10] and the classification method [7] can be used, both of which focus on exploring good text features. The Rule-based methods, such as building a dictionary for each discourse facet [2], are less adaptive. Most studies combine the features of categories with classification algorithms to improve the accuracy of the classification. [2] use a multi-features random forest classifier. The others use a supervised topic model, and XGBOOST [4] and SVM with tf-idf and naive bayes features [6].

Task 2 is a summary task. [11] focus on exploring the sampling process. They use WMD sentence similarity to construct new kernel matrix used in Determinantal Point Processes (DPPs). [4] divide all sentences into three categories (motivations, methods, and conclusions), and then extract sentences from each cluster based on rules and severe features to form a summary. [9] generates a summary by selecting the most relevant sentences from the RP using linguistic and semantic features from RP and CPs. [10] built a summary generation system using the OpenNMT tool.

3 Method

In our approach, we first obtain CTS through feature extraction and content linkage method in the Citation Linage, which is RT (the sentence in RP) related to CT (the sentence in CPs). Then we judge the facet of CTS by feature extraction and classification methods in the facet classification. Finally, a summary of the article is obtained through pre-processing, feature selection, sentence sampling, and post-processing in the summary generation. The framework of our system is shown in Fig. 1.

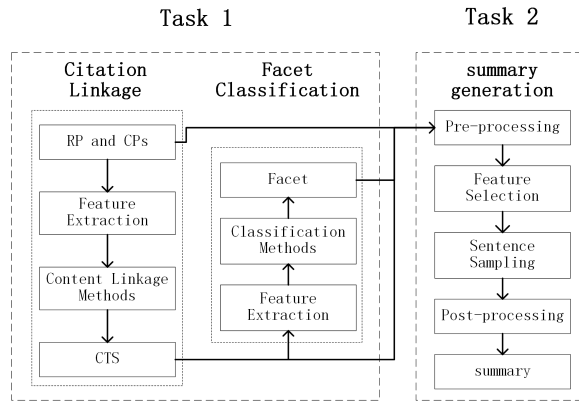


Fig. 1. Framework of our system

3.1 Task 1A

The Citation Linkage task consists of two stages: feature extraction and content Linkage. In feature extraction, we have used some of the good-performing methods of the past, continuing to use word-cos and Word Vector, Sentence similarity (IDF similarity and Jaccard similarity), Context similarities, WordNet. Besides, we add CNN (Convolutional Neural Network) method and LDA-Jaccard. In [11] the LDA vectors of sentences are sparse, that is, the distribution of sentences on topics is sparse. And the LDA vectors pays more attention to whether two sentences belong to the same topic. So we use Jaccard’s idea to express the relativity of the sentence-pair by the ratio of the topic intersection and union of the two sentences, namely LDA-Jaccard.

This paper used Word2vec.H feature as the input of CNN. It is based on word embedding, maps CT and RT information into dense features space, and adds sentence similarity to better guide neural network training. Specifically, CT is represented as an nd matrix CT_Matrix . $CT_Matrix = [wv_1, \dots, wv_i, \dots, wv_n]$. n is the number of words in CT, and d is Word embedding size. wv_i refers to the

word vector of the i -th word in CT. Firstly, we decompose CT_Matrix by SVD to obtain three matrices, U, S, and V. Take the top $\min(n,d)$ values in diagonal of S as the weight set $I_1 = [i_1, i_2, \dots, i_{\min(n,d)}]$, and take the top $\min(n,d)$ rows of V to form CT_V. RT_V and I_2 of RT can also be obtained in the same way. Then the cosine similarity is calculated for each line of CT_V and RT_V to obtain the Word2vec_V. The calculation process is as Fig. 2.

$wv_{i,j} = \text{cosine}(l_i^1, l_j^2)$, l_i^1 and l_j^2 are row vectors of CT_V and RT_V. The cosine similarity is used here.

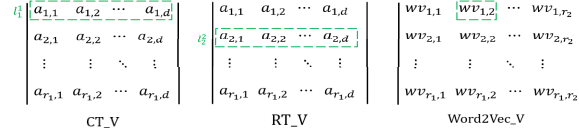


Fig. 2. Word2vec_V building process

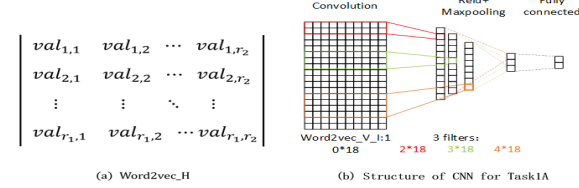


Fig. 3. Word2vec_H and Structure of CNN for Task1A

Finally, we use l_i^1 and l_j^2 to assign weights for rows and columns in Word2vec_V to get the Word2vec_H. $val_{i,j} = i_i^1 \cdot i_j^2$, as shown in Fig. 3(a).

In content linkage, this paper uses the multi-feature fusion method and the binary classification method by CNN. Multi-feature fusion methods include voting1.1, voting2.0, Jaccard-Focused-new, and Jaccard-Cascade. We use the Word2vec_H feature composed of CT and RT as the input of CNN, and the output is the related or unrelated category that CT and RT belong to. The structure of CNN is showed in Fig. 3(b).

3.2 Task 1B

Facet Classification: Our system uses Rule-based methods and Machine Learning Methods for Task 1B. Rule-based methods construct rules based on features extracted from CTS, RP and CPs. According to the results in last year, we only use Subtitle and High Frequency Word Combining Rule (SubHFW) this time. As for Machine Learning methods, we apply Random Forest (RF), a Voting Classifier consisting of 3 Gradient Boosting (GB) and Convolutional Neural Network (CNN) to assign each CTS single or multiple facets. RF and GB take Location of Paragraph, Document Position Ratio, Paragraph Position Ratio and Number of Citations or References as input features while CNN takes the matrix of word embedding of CTS as input. Finally, we combine all the results from Rule-based methods and Machine Learning Methods to obtain a fusion result, which is called Fusion method.

3.3 Task 2

For task2, we would like to present an original Quality-Diversity model for extractive automatic summarization based on the DPP sampling algorithm [12]. In general, a document can be represented as a ground set of items. Each sentence is a minimum item, and the extractive summary can be regarded as a subset from ground set with high quality and low redundancy. Figure 4 shows the framework of our system. The main process for summary generation consists of Pre-processing, Feature Selection, Sentence Sampling and Post-processing.

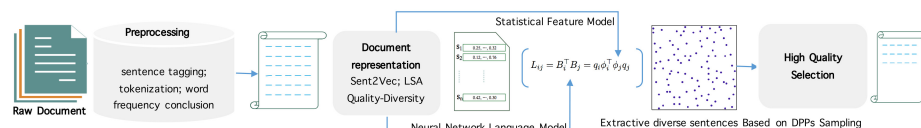


Fig. 4. System Framework for Task 2

Pre-processing First, we need to correct some xml-coding errors manually. Latter, we have to make some preparations such as document merging, sentence filtering and input file generation for hierarchical Latent Dirichlet Allocation (hLDA). We merge the content of RP and the citations into a document for CTS feature described below. Besides, all documents are converted to lowercase letters. Then we filter the corpus for removing some equations, figures, tables and generate input file for hLDA model which contains word index and their corresponding frequencies.

Feature Selection When it comes to document representation, we to build matrix L from both partial (Statistical Feature Method) and holistic (Neural Network Language Model) perspectives to ensure better sentence sampling for summaries. First, we build matrix L through $L_{ij} = q_i S_{ij} q_j$ concretely, we adopt Sentence Length (SL), Sentence Position (SP), Title similarity (TS), CTS, and Hierarchical Topic Model (HTM) as features according to the work of Li L [13] for quality and JACCARD similarity for diversity. We are looking forward to finding a best linear combination of designed qualities in order to capture more obvious characteristic for high-quality summary. Furthermore, we construct matrix L through $L_{ij} = B_i^\top B_j$ by the vectors \mathbf{B} representing sentences from Sent2Vec and LSA directly, and call this framework Neural Network Language Model.

Sentence Sampling We use DPPs to select sentences, which 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 A et al. [12]. We can enhance the diversity of summary by using DPPs. In this way, given the L matrix constructed on document sentences, the sampling method based on DPPs [13] can automatically choose those diverse sentences with high quality as candidate summary sentences.

Post-processing Since we have already had the candidate summary sentences, we can truncate the output summary with sentences ranking high in quality, limit the summary to 250 words, and remove some white spaces in Post-processing.

4 Implementation and Experiments

4.1 Task 1A

In our previous work, we obtained a lot of features. As shown in Table 2, Features number indicates the number of features the method contains. The four methods in [11] have different effects on the test data and the training data, and the more features with good performance are used, the more stable the performance of the testing set is. The more stable the performance is. Therefore, we removed the features with poor performance on the training set, remaining the features with good performance for fusion methods. We adjust the parameters of the four fusion methods in [11]. The four new fusion methods are voting1.2, voting2.1, Jaccard-Focus-1.1, and Jaccard-Cascade-1.1. Since the LDA can discover the topic information and LDA vector is sparse, lexicon (LDA) and LDA-cos are removed and LDA-Jaccard is added. Since the lexicon (co-occurrence) only includes words selected from the training set, when the difference between the testing set and the training set is great, the lexicon (co-occurrence) is ineffective. In the experiment, we chose 600 dimensions for LDA vector and 200 dimensions for word vector. Table 1 shows the parameter settings for our method. As to experiments, we choose 600 dimension as LDA vector and 200 dimension as word vector. Table 1 shows the parameter settings of our methods.

Table 1. Parameter settings of Methods in Task 1A

Feature	V-1.2		V-2.1		J-F-1.1		J-C-1.1	
	W	P	W	P	W	P	W	P
Idf similarity	1	12	0.5	5	0.6	16	0.5	16
Idf context similarity			0.8	3	0.5	15	0.4	10
Jaccard similarity	1	5	0.5	6	JS	7		
Jaccard context similarity			0.5	8	0.7	16	0.6	16
Word vector	1	8	0.5	7	0.5	26		
word-cos	1	10	0.7	7	0.5	26	0.5	10
LDA-Jaccard	1	12	0.4	7				
lin			0.5	5				
jcn					0.6	11		

In addition, with the increasing training data, we begin to try to solve task1A with CNN. In this paper, we build the Word2vec_H feature for the sentence-pair, so that we could reduce the dimensionality of the input and add the cosine similarity to it. We use V-1.2, V-2.1, J-F-1.1, J-C-1.1, and W_H-C to represent Voting-1.1, Voting-2.0, JacCard-Focused-1.1, and Jaccard-Cascade-1.1,

Word2vec.H-CNN respectively. In Table 1, W and P are Weight and Proportion respectively. JS means 10 fold of Jaccard Similarity.

Table 2. Performance of Methods in Task 1A in 2018

Method	F1-train	F1-test	(F1-train)-(F1-test)	Features number
V-1.1	0.147	0.113	0.034	4
V-2.0	0.128	0.122	0.006	7
J-F	0.132	0.114	0.018	8
J-C	0.116	0.09	0.026	4

According to Table 2 we predict that V-1.2 and J-F-1.1 will be more stable on the testing set. The W.H-C uses the data in "Training-Set-2019", and the effect is the worst due to some problems, such as data imbalance of training set and complex structure of CNN.

Table 3. Performance of Methods in Task 1A in 2019

Method	F1-train	F1-test	(F1-train)-(F1-test)	Features number
V-1.2	0.097	0.106	0.007	5
V-2.1	0.105	0.104	0.001	8
J-F-1.1	0.105	0.103	0.002	7
J-C-1.1	0.099	0.087	0.026	4

From Table 2 and Table 3 [14], we can get three conclusions:

The number of features used in V-1.2 is less than V-2.1 and J-F-1.1, but the result of V-1.2 is similar to V-2.1 and J-F-1.1. The number of features used in V-1.2 is about the same with J-C-1.1, and the result of V-1.2 is better than J-C-1.1. It shows that features used in V-1.2 play a leading role.

The results of the runs in 2019 verify our prediction, that is, the more features that are used, the more stable the performance on the test set is. So the performance on the testing set and the training set of V-2.1 is very stable, as well as J-F-1.1.

After removing co-occurrence dictionary, (F-train) - (F-test) results are smaller, which indicates that co-occurrence dictionary has limitations and should be removed.

4.2 Task 1B

In this section, we will introduce our methods applied for Task 1B in detail.

Rule-based Methods: Subtitle Rule: We use subtitles of CTS and citance to determine which facet they belong to. If subtitles contain five predefined classes,

we categorize CTS and citance to corresponding facet. High Frequency Word Rule: We use high frequency words of each class to classify CTS and citance. We first remove common words, and then set a threshold for each facet. Subtitle and High Frequency Word Combining Rule: We first apply Subtitle Rule to obtain the facet. If it doesn't give an explicit answer, then we use High Frequency Words Rule to obtain facet.

Machine Learning Methods: Firstly, we extract features from CTS and citance consisting of Location of Paragraph, Document Position Ratio, Paragraph Position Ratio and Number of Citations or References and concatenate these features into an 8-dimension vector. Then we train RF and GB based on the features. As for CNN, the content of CTS is transformed to a matrix where i th row corresponds to the word embedding of i th word and j th column represents the j th dimensionality of the embedding. Then, we stack a convolutional layer with multiply kernel sizes followed by a max-pooling layer. The architecture of CNN is shown in Fig 5.

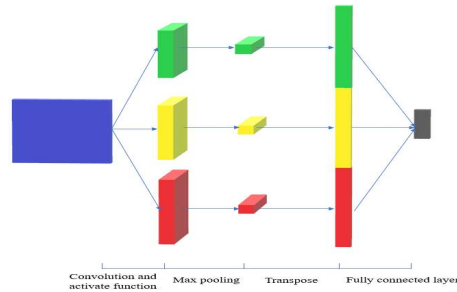


Fig. 5. Architecture of CNN for Task 1B

Table 4. Results in 2019

Method	Train-set(F1 Score)	Test-set(F1 Score)
RF	0.3281	-
SubHFW	0.3556	0.389
Voting	0.3611	0.341
CNN	0.2841	0.342

Results on Train-Set-2019 are illustrated in Table 4. We find that Voting and SubHFW methods have a better performance. CNN performs poorer than we expected since the training data set is too small for a neural network to learn. And the dataset is imbalanced where method facet has more samples than other facets.

As for Task 1B, the results on Test-set-2019 show that SubHFW method outperforms than other method and ranks first among all methods, which indicates that the features of subtitle and high frequency word are crucial to determine

the facet of each CTS. Moreover, textCNN method performs poorer than we expected due to the demand of larger dataset.

4.3 Task 2

The results below utilize Manual ROUGE values to evaluate our system summary. During the evaluation phase, CL-SciSumm 2018 has provided THREE kinds of criterion for option: the collection of citation sentences (the community summary), faceted summaries of the traditional self-summary (the abstract), and ones written by well-trained annotators (the human summary).

Take community summary for instance, we test each feature SP (φ_0), SL (φ_1), TS (φ_2), HTM (φ_3), and CTS (φ_4) described in subsection 3.3 on statistical fearture model independently to figure out its own contribution at first. As the CTS feature (φ_4) is specially designed, we tend not to present its individual performance, but record and observe the binary combination with every other basic feature.

Table 5. Binary Combination Test on Quality

Run ID	φ_0	φ_1	φ_2	φ_3	φ_4	ROUGE1	ROUGE2
quality-0	1	0	0	0	1	0.43652	0.23824
quality-1	0	1	0	0	1	0.42104	0.19574
quality-2	0	0	1	0	1	0.52800	0.37682
quality-3	0	0	0	1	1	0.41193	0.18440

From TABLE 5, the best binary combination comes from TS (φ_2) and CTS (φ_4) features. One possible explanation is that the community summary itself has already included these citation sentences. With the title containing the essence of a paper, selected sentences following this ranking rule will definitely guarantee the overlapping on golden summaries.

Table 6. Statistical Feature Model Performance on Community Summary

Run ID	φ_0	φ_1	φ_2	φ_3	φ_4	ROUGE1	ROUGE2
Statistical-0	0	0	1	0	1	0.52552	0.37333
Statistical-1	0	0	0	1	1	0.41283	0.18438
Statistical-2	1	0	2	0	2	0.44104	0.24653
Statistical-3	1	1	2	0	2	0.42219	0.22141

Analogically, we conduct experiments on other two kinds of golden summaries, where the weights of parameters appear slightly different. From TABLE 6 and TABLE 7, which present the results of the community summary and human summary separately: the best binary combination goes to the same tendency. The phenomena of same best combination maybe interpreted implicitly that no

matter whether the sentences are cited otherwise or the summaries are written by annotators, they two both are from the perspective of readers. Community summaries consist of those citation sentences, and the sentences themselves are extracted from the original documents, thus there’s no wonder the ROUGE evaluation is far higher than other kinds of summaries. However, human summary is based on comprehension of readers. In this case we do extra experiments on human summaries besides the same parameter setting as community summaries. The best new combination as TABLE 7 shows is a little bit different from the previous mere copies of community summaries. When it comes to human summary, the more parameters are involved, the higher ROUGE F-score it reaches. Unfortunately, for community summary, when we desire a further exploration on binary combination, any additional attribute performs adversely. *There are a thousand Hamlets in a thousand people’s eyes.*

Table 7. Statistical Feature Model Performance on Human Summary

Run ID	φ_0	φ_1	φ_2	φ_3	φ_4	ROUGE1	ROUGE2
Statistical-0	0	0	1	0	1	0.41900	0.19167
Statistical-1	1	0	0	0	1	0.39866	0.15321
Statistical-2	2	0	3	0	3	0.43504	0.25430
Statistical-3	2	1	3	0	3	0.42219	0.22141

Table 8. Statistical Feature Model Performance on Self-summary

Run ID	φ_0	φ_1	φ_2	φ_3	φ_4	ROUGE1	ROUGE2
Statistical-0	1	0	0	0	0	0.39630	0.19019
Statistical-1	0	1	0	0	0	0.35020	0.11507
Statistical-2	0	0	1	0	0	0.38434	0.17296
Statistical-3	0	0	0	1	0	0.30237	0.08561
Statistical-4	2	0	3	0	3	0.42688	0.37234

As for the self-summary (the abstract), things presented in TABLE 8 are opposite. Every binary combination with CTS (φ_4) feature are not that satisfied, so we present each individual contribution of other statistical or topic features. Also, we try the best parameter setting for community summary and human summary both on abstract summary. Perhaps although we have tried our best to follow the writers, there always exists a narrow gap between our readers’ comprehension and writers’ original intention. This part of the experiment follows a simple but practical principle that on the condition that we cannot fully understand latent semantics the writers want to express, we still manage to deal with some statistical features which help to extract important sentences. If the summarizer is developed through this approach, it is not limited in a familiar language and does not require any additional linguistic knowledge or complex linguistic processing.

Furthermore, when extracting sentences from the Neural Network Language Model (using Sent2Vec/LSA representation for sentences), we choose the best

quality combination for community summary, human summary and abstract summary. TABLE 9 suggests the Neural Network Language Model performance. Besides, TABLE 10 shows the best results of several runs in BIRNDL 2019. Among all the systems in competition, our system won the first prize for the human summary, while the second place for abstract and community summary.

Table 9. Neural Network Language Model Performance

Run ID	φ_0	φ_1	φ_2	φ_3	φ_4	ROUGE1	ROUGE2
community-Sent2Vec	0	0	1	0	1	0.52254	0.31893
human-Sent2Vec	2	0	3	0	3	0.41716	0.18210
abstract-Sent2Vec	2	0	3	0	3	0.42114	0.21823
community-LSA-1	0	0	1	0	1	0.56971	0.44228
community-LSA-2	0	0	1	0	1	0.59240	0.46528
human-LSA-1	2	0	3	0	3	0.37717	0.17825
human-LSA-2	2	0	3	0	3	0.38568	0.17424
abstract-LSA-1	2	0	3	0	3	0.39944	0.20009
abstract-LSA-2	2	0	3	0	3	0.40051	0.18617

Table 10. Best Results on BIRNDL 2019

Run ID	Abstract R2	Abstract SU4	Community R2	Community SU4	Human R2	Human SU4
run3	0.389	0.210	0.122	0.063	0.278	0.200
run19	0.386	0.227	0.121	0.063	0.257	0.189
run15	0.381	0.211	0.119	0.062	0.267	0.191

5 Conclusion and Future Work

This year, we have added neural networks to the methods of three tasks. We hope to make use of large training corpus to give the advantages of neural networks, that is, deeply mining the meaning of the text. Rule-based and statistics-based methods have achieved good performance, so we try to combine them with neural networks. In the future work, Task 1A is expected to automatically adjust the weight of features through neural network and combine multiple features better. For Task 1B, more study should be done to reduce the impact of imbalanced data on neural networks. Besides, more crucial features are expected to be found since the performance of machine learning methods is the best so far. In Task 2, we expect the neural network language models to make contributions to obtain more meaningful semantic representation for sentences against statistical features.

Acknowledgements

This work was supported in part by the Beijing Municipal Commission of Science and Technology under Grant Z181100001018035; National Social Science

Foundation of China under Grant 16ZDA055; National Natural Science Foundation of China under Grant 91546121; Engineering Research Center of Information Networks, Ministry of Education.

References

1. CL-SciSumm 2019 Homepage, <http://wing.comp.nus.edu.sg/cl-scisumm2019/>.
2. Wang P, Li S, Wang T, et al. NU DT@ CLSciSumm-18In: Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries[C]//BIRNDL@ SIGIR. 2018: 102-113.
3. Davoodi E, Madan K, Gu J. CLSciSumm Shared Task: On the Contribution of Similarity measure and Natural Language Processing Features for Citing Problem[C]//BIRNDL@ SIGIR. 2018: 96-101.
4. Ma S, Zhang H, Xu J, et al. NJUST@ CLSciSumm-18[C]//BIRNDL@ SIGIR. 2018: 114-129.
5. Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.
6. Agrawal K, Mittal A. IIT-H@ CLScisumm-18 In: Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries[C]//BIRNDL@ SIGIR. 2018: 130-133.
7. Baruah G, Kolla M. Klick Labs at CL-SciSumm 2018[C]//BIRNDL@ SIGIR. 2018: 134-141.
8. Karimi S, Moraes L F T, Das A, et al. University of Houston@ CL-SciSumm 2017: Positional language Models, Structural Correspondence Learning and Textual Entailment[C]//BIRNDL@ SIGIR (2). 2017: 73-85.
9. Aburaed A, Bravo A, Chiruzzo L, et al. LaSTUS/TALN+ INCO@ CL-SciSumm 2018-Using Regression and Convolutions for Cross-document Semantic Linking and Summarization of Scholarly Literature[C]//Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL2018). Ann Arbor, Michigan (July 2018). 2018.
10. Debnath D, Achom A, Pakray P. NLP-NITMZ@ CLScisumm-18 In: Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries[C]//BIRNDL@ SIGIR. 2018: 164-171.
11. Li L, Chi J, Chen M, et al. CIST@ CLSciSumm-18: Methods for Computational Linguistics Scientific Citation Linkage, Facet Classification and Summarization[C]//BIRNDL@ SIGIR. 2018: 84-95.
12. Alex Kulesza and Ben Taskar (2012), Determinantal Point Processes for Machine Learning, Foundations and Trends in Machine Learning: Vol. 5: No. 23, pp 123- 286. <http://dx.doi.org/10.1561/22000000044>.
13. Li L, Zhang Y, Chi J et al. UIDS: A Multilingual Document Summarization Framework Based on Summary Diversity and Hierarchical Topics [M] // Li L, Zhang Y, Chi J et al. Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, 2017: 2017: 343-354.
14. Chandrasekaran, M.K., Yasunaga, M., Radev, D., Freitag, D., Kan, M.-Y. "Overview and Results: CL-SciSumm SharedTask 2019", In Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) @ SIGIR 2019, Paris, France.