NJU@CL-SciSumm-19

Hyonil Kim[1,3][0000-0002-6380-9507] and Shiyan Ou[1,2] [0000-0001-8617-6987]

[1] School of Information Management, Nanjing University, Nanjing, China
[2] oushiyan@nju.edu.cn
[3] kimhyonil@126.com

Abstract. Cited text identification is helpful for meaningful scientific literature summarization. In this paper, we introduces our system submitted to the CL-SciSumm 2019 Shared Task 1A. Our system have two stages: similarity-based ranking and supervised listwise ranking. Firstly, we select the top-5 sentences per a citation text, due to the modified Jaccard similarity. Secondly, these top-5 selected sentences are proceeded to rank by a CiteListNet (listwise ranking model based on deep learning). Our experiments showed that our proposed method outperformed other prior methods on the CL-SciSumm 2017 test dataset.

Keywords: Cited Text Identification, Cited Text, Listwise Ranking, Citation Content Analysis, Text Similarity.

1 INTRODUCTION

Automatic summarization of academic paper may be a very effective solution to avoid the information overload of researchers and to understand the state-of-the-art of the research topic.

The CL-SciSumm shared tasks explore the solutions for the making a comprehensible summary of an academic paper given its citation text. These tasks focuses on the sentence-level cited text information to perform the summarization of a paper. To this end, the identification of the cited text should be done. CL-SciSumm Shared Task 1A is just to identify the spans of cited text in reference paper (RP) that contain the given citation text of its citing paper (CP).

In this paper, we use various similarity metrics to evaluate the similarity between a citation text and its candidate cited sentence, and adopt listwise ranking algorithm to train our ranking model.

2 RELATED WORK

Most of previous studies regarded the identification of cited texts as a classification problem and thus used some machine learning algorithms like SVM, Random Forest, CNN to train text classifiers. To build the classifiers, various features were explored by researchers. Ma et al. (2018) chose Jaccard similarity, cosine similarity and some position information as features, and trained four classifiers including Decision Tree, Logistic Regression and SVM.[9] Finally they used a weighted voting method to combine the categorization results of the four classifiers and achieved the best performance in CL-SciSumm 2017 competition. Yeh et al. (2017) considered some lexical features, knowledge-based features, corpus-based features, syntactic features, surface features to represent the feature vector and adopted a majority voting method to combine the results of the six classifiers like KNN, Decision Tree, Logistic Regression, Naive Bayes, SVM and Random Forest. They got the F value of 14.9% by running their system on the corpus of the CL-SciSumm 2016 competition.[3]

There are two main issues in the categorization-based methods: local ranking and class-imbalanced data. On the one hand, the cited text identification problem should be regarded as a ranking problem rather than a classification one because we only intent to choose the sentence(s) that contains more similar content with the citation sentence(s) compared to other sentences. On the other hand, there is only few sentences (usually not more than five) to be cited sentences in a target paper. Sometimes the ratio of the negative and positive sample in a corpus is even greater than 150. Ma et al. (2018) used Nearest Neighbor (NN) rule (Wilson, 1972) to reduce data imbalance and increased the F1-score from 11.8% to 12.5%.[9]

With respect to the ranking-based cited text identification, a few studies have been done. Dipankar et al. (2017) ranked the sentences in a target paper according to the cosine similarity between each candidate sentence and the citation sentences to select the top five sentences as the cited sentences.[2] However, this unsupervised method did not obtain reasonable performance. Therefore, we proposed a listwise ranking method for identifying cited sentences, which is supervised method trained by a deep learning mechanism.

3 Methodology

In this study, we regarded cited text identification as a ranking problem, and proposed a ranking-based method to identify citation sentences based on deep learning. This method includes two stages of ranking: a similarity-based unsupervised ranking and a supervised listwise ranking. Since the cited text was deemed to contain more similar content with the cited text than other sentences in the same paper, we first ranked all the sentences in a reference paper according to each sentence's similarity with a cited text. Then we choose top K sentences to create a subset of the given train corpus for the second stage ranking, while the Kth sentence obtained the best F-value according to the given training corpus. In the second stage, a listwise ranking model was trained on the subset training corpus to rank the K sentences and then top N sentences (N<K) were selected as the cited sentences.

3.1 Similarity-based Ranking

In this section, we explored different similarity metrics between two texts to rank all the sentences in a cited text (named as candidate sentences) for a specific citation text in one of its citing paper, whereas a citation text may contain one or more sentences. We considered five kinds of similarity metrics, including TFIDF-based cosine similarity, word embedding-based cosine similarity, SVM Kernel functions-based cosine similarity, Jaccard-like similarity and BM25.

Cosine Similarity based on TFIDF Weighted Vector Space Model. In this kind of similarity metrics, N-grams were extracted from a text as its features and represented the text as a TFIDF weighted feature vector based on Vector Space Model and calculated the cosine similarity between two feature vectors. We tested two TFIDF-based cosine similarity metrics: when N=1 and when N=2 respectively for N-grams.

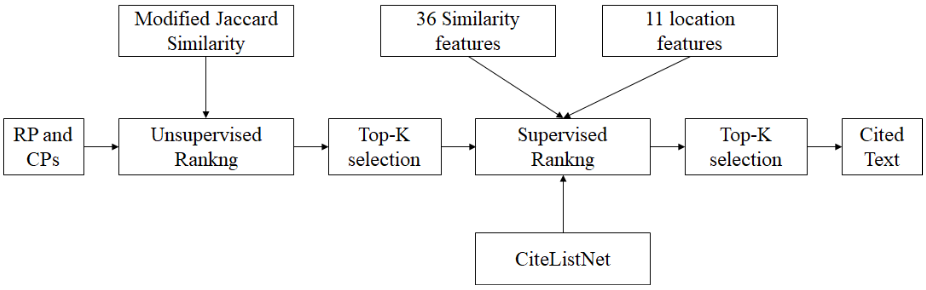

Fig. 1. Two-stage ranking for the identification of cited text.

Cosine Similarity based on Word Embedding. In this kind of similarity metrics, a text was represented as a text vector based on word embedding. First, the vector of each word in a text was trained with Word2vec. Then three kinds of text vectors were calculated based on the word vectors using different weighting mechanisms, TF-weighted average of the word vectors (TF-AWV), TFIDF-weighted average (TFIDF-AWV), and T-statistics-weighted average (Tstat-AWV). Here, the T-statistic of a word refers to the T-test statistics of the hypothesis that whether the word appears in a text or not is independent to whether or not the text is a cited text.

Cosine Similarity based on SVM kernel functions. As we know, a linear inseparable sample can become linearly separable by projecting a low-dimensional space to a high-dimensional one. Thus we considered to transform the TFIDF-weighted average word vector (TFIDF-AWV) to a higher-dimensional space by using three SVM kernel functions, so as to make a text more distinguishable from others. Thus three kinds of kernel functions-based cosine similarity were calculated with the following equation based on three kernel functions respectively, i.e. 2-dimension polynomial, 3-dimension polynomial function and RBF function.

$$cos(\phi(A),\phi(B)) = \frac{\phi(A)\cdot\phi(B)}{\sqrt{\phi(A)\cdot\phi(A)}\cdot\sqrt{\phi(B)\cdot\phi(B)}} = \frac{K(A,B)}{\sqrt{K(A,A)}\cdot\sqrt{K(B,B)}} \quad (1)$$

Where $\phi(\cdot)$ denotes a mapping function, by which a vector space can be mapped to another space, $K(\cdot,\cdot)$ refers to a kernel function that follows Mercer's condition[10], and A (or B) refers to the vectors of the original space.

Cosine Similarity based on SVM Kernel Functions. As we know, a linear inseparable sample can become linearly separable by projecting a low-dimensional space to a high-dimensional one. Thus we considered to transform the TFIDF-weighted average word vector (TFIDF-AWV[11]) to a higher-dimensional space by using three SVM kernel functions, so as to make a text more distinguishable from others. Thus three kinds of kernel functions-based cosine similarity were calculated with the following equation based on three kernel functions respectively, i.e. 2-dimension polynomial, 3-dimension polynomial function and RBF function.

Jaccard-like Similarity. A citation text in a citing paper and each candidate sentence in a reference paper can be regarded respectively as a set of N-grams (N=1, 2, 3). We used some Jaccard-like similarities to calculate the overlap between the N-gram set of the citation text and the one of each candidate cited sentence. In addition to the standard Jaccard similarity metric, two variations, MJS1 (see eq. 3) and MJS2 (see eq. 4) were also considered.

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (2)$$

$$MJS1(A,B) = \frac{|A \cap B|}{|A|} \quad (3)$$

$$MJS2(A,B) = \frac{|A \cap B|}{|\mathrm{B}|} \quad (4)$$

Additionally, there are two different applicable weighting methods, respectively IDF-weighting and T-statistic weighting, to improve Jaccard similarity metric.

B25 measure. If we make a comparison of cited text identification issue with information retrieval, citation text may correspond to query, while cited sentence to document to be retrieved. Since B25 measure is well-known metric in classical information retrieval, it may also serve as similarity metric in our work.

In total, there were 36 similarity metrics (as shown in Table 1), which were used in the first-stage similarity ranking.

Table 1. The similarity metrics used in the similarity-based ranking.

Similarity metrics	Subclass	Number of features
Cosine similarity metric	TF-IDF based VSM	2
	Word2vec based model	3
	SVM Kernel transforming model	3
Jaccard similarity metric	Unigram	9
	Bigram	9
	Trigram	9
BM25		1

Total		36

3.2 Top-K sentence selection based on similarity ranking

Among above 36 similarity metrics, we try to select the best metric as the top-k selection feature. We used F1.5 measure（β=1.5）to evaluate which similarity metric work well, since the top K sentence selection result are expected that it involves the more positive samples.

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R} \tag{5}$$

We used the training data of the CL-SciSumm 2017 dataset to evaluate the performance of each kind of similarity metric. We found that the 33th feature (T-statistic weighted tri-gram MJS1 similarity) is best metric for top-K selection in the first stage. Table 2 shows the performance of the top-k selection based on the 33th similarity metric.

Table 2. The performance of top-K sentence selection based on Jaccard similarity on the training data of CL-SciSumm 2017 datasets.

K	Recall	Precision	$F_{1.5}$-measure
1	9.96	14.35	10.99
2	16.17	11.65	14.45
3	21.79	10.46	16.35
4	25.74	9.27	16.64
5	29.39	8.47	**16.69**
6	32.94	7.91	16.69
7	35.40	7.28	16.18
8	36.98	6.66	15.40

From the result like as shown in Table 2, we selected top-5 selection based on T-statistic weighted tri-gram MJS1 similarity, which was also used as a baseline system.

3.3 Supervised Listwise ranking based on NN model

Recently, deep learning approaches have gradually gained popularity in artificial intelligent problems. However, it is usually applied in classification problem, but not in ranking problem.

For ranking, there are three kinds of methods: pointwise, pairwise and listwise. Pairwise ranking is almost same effect with classification, in other words, this task is formalized as classification of object pairs (correct sample and incorrect sample), such as RankSVM [4, 7] and RankNet [1]. Pairwise ranking can be referred as multiple classification, therefore, its performance can be depend on the individual classification, however, in cited text identification, the performance of the individual classifier is not ideal. In contrast, listwise ranking approach directly consider overall rank of all samples of the list, but it is needed to annotate the order of all samples. In the cited text identification task, however, it is very difficult to annotate all the order of samples.

Therefore, we proposed a novel listwise ranking model, namely CiteListNet, based on deep learnings. Given citance c and reference sentence list $r_1, \ldots, r_n$, the our task can be formulated as following:

$$S(c_i, r_{ij}) = Softmax(F(c_i, r_{ij})) \ = \frac{\exp(F(c_i, r_{ij}))}{\sum_j \exp\left(F(c_i, r_{ij})\right)} \tag{6}$$

$$Obejctive\text{: } \max \textstyle\sum_{i,j} y_{ij} * S(c_i, r_{ij}) \tag{7}$$

$$y_{ij} = \begin{cases} 1, \ r_{ij} \ is \ cited \ text \ of \ a \ citaion \ c_i \\ 0, \ r_{ij} \ is \ not \ cited \ text \ of \ a \ citation \ c_i \end{cases} \tag{8}$$

Where F(.) denotes a deep learning model, which can be implemented as any architecture.

In above optimization, we only consider the order of positive sample, but not negative sample. The more the score of positive sample is, the higher the order of it is. If its score is over 0.5, then its order become the first.

In addition to similarity-based metric, we also involved the section information in our feature set. Section information represents in which kind of section the reference sentence appears in the reference paper. The kind of section is decided based on rule-based method. Table 3 shows the kinds of section which frequently appear in the computational linguistics field.

Table 3. The various kinds of section.

No	Kind of section	No	Kind of section
1	Title	7	Analysis
2	Abstract	8	Experiment
3	Introduction	9	Data
4	Related work	10	Future work
5	Method	11	Other
6	Conclusion		

4 Experiments and Result

The CL-SciSumm 2017 datasets contain 30 topics as training data and 10 topics as test data [5], where each topic consists of a reference paper and some citing papers that involves the citation to the reference paper. In this section, we will exploit these datasets to demonstrate our proposed method.

To train the list-wise ranking model, we prepared the subset from the training data through the top-5 sentence selection as described in Section 3.2.

In our experiment, the top-2 listwise ranking showed the best performance on the training data where its F1-score is observed as 17.9%.

In order to validate the generalization ability of our method, we try to evaluate the trained model on the test data in the CL-SciSumm 2017 datasets. The result demonstrated that the top-2 listwise ranking showed a best performance and that the overfitting did not occur during our listwise ranking training.

We also compared our method with earlier approaches as shown in Table 4. To evaluate the performance of the different systems, two kinds of metric is used: sentence ID overlap and ROUGE scoring [5, 6, 8]. The former use the raw number of overlapping sentences between system output and the gold standard to calculate the precision, recall and F1-score. This evaluation also exploit the micro-average and macro-average respectively. As shown in Table 4, our proposed listwise rank model showed the best performance over all the kinds of evaluation metric. Furthermore, baseline (MJS-based top-5 sentence selection) also showed the comparable performance.

The experiment results demonstrate that our proposed method outperforms any prior approaches.

Table 4. The performances of various system on the test data in CL-SciSumm 2017 Shared Task[1].

System	Micro-Avg (F_1)	Macro-Avg (F_1)	ROUGE-2 (F_1)
NJUST	12.3	14.6	11.4
TUGRAZ	11.0	13.5	10.8
CIST	10.7	11.3	4.7
NUDT	14.8	-	-
CiteListNet	**15.3**	**18.3**	**14.3**
Baseline	11.7	12.9	4.9

As shown in Table 5, top-2 ranking also showed a good performance on the test data in CL-SciSumm 19 Shared Task.

Table 5. The performances of various top-N ranking on the test data in CL-SciSumm 2019 Shared Task [12].

Top-N	Micro-Avg (F_1)	ROUGE-SU4 (F_1)
2	12.4	9.0
3	11.8	7.9
4	10.4	4.1
5 (Baseline)	9.8	3.0

[1] In Table 4, the result of the prior works is from Jaidka's report [5].

5 Discussion

In this paper, we focused on the cited text identification issue and proposed a novel method, namely CiteListNet, based on listwise ranking.

The main contributions of this paper are two points: feature selection and a novel listwise ranking model.

First, we adopted new features to identify cited text. We modified Jaccard similarity to consider how much the cited sentence covers the citation text and how much the citation text covers the cited text. The former is verified to be useful for identifying cited text, while the letter is less helpful than other features. When using the Jaccard similarity, it is recommended to use N-gram language model. Moreover, we found that T-statistics that represents how much the word is probable to appear in the cited text could be used as a useful weight. The experimental result shows that T-statistics weighted Modified Jaccard Similarity feature based on tri-gram language model is the most useful feature and MJS-based top-5 sentence selection shows the comparable performance, although no training is done.

Second, we proposed a novel listwise ranking model based on deep learning - CiteListNet. We found that our proposed method is stable and did not occur overfitting problem during the training process. The result of our experiment shows that our novel method outperforms other prior approaches.

In this paper, it was still not considered that cited text could be represented as a paragraph. In this case, the relationship between the cited sentence and the citation text may have a different characteristic. In the future work, we will focus this issue, and apply this research result to various bibliometric task.

Acknowledgement. This paper is one of the research outputs of the project supported by the State Key Program of National Social Science Foundation of China (Grant No. 17ATQ001).

References

1. Burges, C., Shaked, T., Renshaw, E., et al.: Learning to rank using gradient descent. In: Proceedings of ICML 2005, pp. 89-96. (2005)
2. Dipankar Das, S.M., Pramanick, A.: Employing Word Vectors for Identifying, Classifying and Summarizing Scientific Documents. In: Proc. of the 2nd Joint Workshop on Bibliometric-enhanced Inforamtion Retrieval and Natural Language Processing for Digital Libraries (BIRND 2017). Tokyo (2017)
3. Yeh J.Y., Hsu T.Y., Tsai C.J., et al.: On identifying Cited Texts for Citances and Classifying Their Discourse Facets by Classification Techniques. Journal of Information Science and Engineering, Vol. 35(1), 61-86 (2016)
4. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Proceedings of ICANN 1999, pp. 97-102. (1999)
5. Jaidka, K., Chandrasekaran, M.K., Jain, D., et al.: The CL-SciSumm Shared Task 2017: Results and Key Insights. In: Proc. of the 2nd Joint Workshop on Bibliometric-enhanced

Information Retrieval and Natural Language Processing for Digital Libraries (BIRND 2017), Tokyo (2017)

6. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., et al.: Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. International Journal on Digital Libraries, Vol. 19(2-3), 163-171 (2018)
7. Lee, C.P., Lin C.J.: Large-Scale Linear RankSVM, Neural Computation, Vol. 26, 781-817 (2014)
8. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004). 2004.
9. Ma, S., Xu, J., Zhang, C.: Automatic identification of cited text spans: a multi-classifier approach over imbalanced dataset. Scientometrics, Vol. 116, 1303-1330. (2018)
10. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society A, Vol. 209(441-458), 415-446. (1909)
11. Ou, S.Y., Kim, H.I.:Unsupervised Citation Sentence Identification Based on Similarity Measurement. In: International Conference on Information. Springer, Cham (2018)
12. 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.