NLP-PINGAN-TECH @ CL-SciSumm 2020

Ling Chai, Guizhen Fu, Yuan Ni
PingAn Health Technology, Shenzhen, China
{CHAILING123, FUGUIZHEN037, NIYUAN442}@pingan.com.cn

Abstract

CL-SciSumm Shared Task at EMNLP 2020 Workshop consists of three subtasks about automatic summarization for research papers. This paper introduces the systems of Task 1A and Task 1B submitted by team NLP-PINGAN-TECH. TASK1A is to identify the cited text spans in the reference paper, and Task 1B is to determine the discourse facet of the cited text. Task 1A is regarded as a binary classification task of sentence pairs and the strategies based on language models are proposed. Integration with contextualized embedding with extra information is further explored in this article. For Task 1B, the pre-trained language models are fine-tuned to accomplish a multi-label classification task. The results show that extra information can improve the identification of cited text spans. The end-to-end trained models outperform models trained with two stages, and the averaged prediction of multi-models is more accurate than an individual one.

1 Introduction

With the ever-increasing scientific publications, tracking research status from an extremely huge amount of research papers is getting harder for scholars. To solve this problem, research on automatic summarization provides an efficient way to get the highlights of articles for readers. Citation-based summarization methods leverage information of citing and cited texts to construct the summary of the reference paper (Abu-Jbara, Amjad et al., 1981). Citation texts are considered to contain the most valuable parts for researchers to follow. Moreover, different citing text spans form a relatively complete figure of an article, such as hypothesis, methods, results, and so on. Based on the aforementioned ideas, the CL-SciSumm Shared Tasks propose different tasks about different aspects for scientific publication summarization systems from 2016 to this year (Chandrasekaran, Muthu et al., 2019).

In more detail, the CL-SciSumm 2020 is organized as follows (Chandrasekaran, Muthu et al., 2020).

A set of reference papers (RP) and the citing papers (CPs) that all contain citations to the RP are given. In each CP, the text spans of each particular citation to the RP should be identified. The three subtasks are shown below:

Task 1A: For each citance, identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. These are of the granularity of a sentence fragment.

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 task): Finally, generate a structured summary of the RP from the cited text spans of the RP.

We focus on systems for **Task 1 (Task 1A and Task 1B)** in this paper. Task 1 are evaluated by Sentence Overlap scores and ROUGE-SU4 scores (Chandrasekaran, Muthu et al., 2020).

2 Data Pre-processing

The training sets consist of two datasets: one is 40 manually annotated reference papers and

Proceedings of the First Workshop on Scholarly Document Processing, pages 235–241

https://doi.org/10.18653/v1/P17

Dataset	Auto-annotated	Manually-annotated
total citance	14903	537
cited sentences(>1)	14885	196
consecutive cited sentence	724	75

Table 1: Distribution of cited text

Dataset	Auto-annotated	Manually-annotated
total cited sentence	19869	522
cited only once	14834	370
cited more than twice	5035	152

Table 2: Cited frequency

their citances, the other is 1000 documents auto-annotated by ScisummNet (Nomoto, T. et al., 2018).

We clean and filter out the noisy sentence using NLTK tools (Bird et al., 2009). We then concatenate the sentences badly segmented, like "this is the first part of a sentence" and "(xx and xx, 2019)". The sentences with less than 4 tokens are eliminated. The citing or cited text spans with more than one sentence have been split into multiple citances. As a result, the processed data are all "sentence-sentence" pairs.

Firstly, we compare the two datasets on the distribution of cited sentences and cited frequency which is shown in Table 1 and Table 2. It is clear that manually-annotated data have less cited text spans with more than one sentence and multiple sentences are more likely consecutive, compared to the auto-annotated data. In the meanwhile, the two datasets have a similar repetition rate of cited sentences. We also tried to train models with the manually-annotated dataset mixed with auto-annotated data, but the results demonstrate that the 1000-document dataset generally has negative effects. Therefore, in the following part, we only describe the systems trained on the 40-document dataset. 32 of 40 reference papers are regarded as train data and the rest 8 papers are development set. To reduce the impact of data imbalance, we randomly select 4 negative sentences for each piece of citance for the trainset and do nothing on the development set.

3 Method

3.1 TASK 1A

We consider three kinds of approaches for the identification of cited text spans, both based on the concept of identifying sentence relevance between the citing and cited text spans. All the methods for Task1A are binary classification. For the development set, the two or three sentences with the highest scores for one citing sentence are selected as its corresponding cited text spans. We refer to the citing-cited sentences as sentence A and sentence B in the following sections.

BERT-based methods:

For the first approach, we explore the methods based on the BERT framework (Devlin, Jacob et al., 2018). We use the BERT-base-uncased model as the baseline and then find that using domain-specific embedding and extra information utilization can both improve the performance.

Domain-specific embeddings. It has been proven that domain-specific text embedding could better interpret the semantic knowledge of text spans (Beltagy, Iz et al., 2019). To construct embeddings of texts in the scientific research domain, we propose two approaches:

1. To leverage language model SciBERT (Beltagy, Iz et al., 2019). SciBERT was pre-trained on more than 1,14 million scientific publications (82% on biomedicine and 12% on computer science).
2. To fine-tune the BERT model with scientific documents. Considering that train data are all computational linguistics scientific documents, we feed ACL anthology reference corpus into the BERT-Large-Uncased model (BERT-large has 24 layers, 1024 hidden size, 16 self-attention heads, which total has 340 million parameters.) and train it by running both Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks as the

original paper did (Devlin, Jacob et al., 2018). We refer to this model as ACLBERT.
We fine tune the two aforementioned models on our trainset with learning rate of 5e-5 and with 10% training steps as warm up stage. It's worth noting that we feed sentence pairs in the form of "[CLS] sentence A [SEP] sentence B [SEP]" into models, then we also train another model with order the of sentence A and sentence B reversed. The averaged prediction of two models with different sentence orders is better than that of a single model. The final submitted systems SciBERT and ACLBERT in Table 3 are all the averaged predictions of two models as described above.

Extra information utilization. To capture the features at the document level, we try to add the position and section features into the whole model. Position information "sid" and "ssid" (index of the entire document and section, respectively) are already given. With the full text of the reference paper given, we parse the documents and rearrange all section expression to nine categories: title, abstract, introduction, related work, method, experiment, result, conclusion, and none (for texts without sections).

Firstly, we add those three features as the prefixes "[method] [sid=xx] [ssid =xx] " for all sentence B. The section text like "method" and position information like "23" will be treated as normal tokens, and the characters "[" and "]" will be identified to token "[UNK]" but have the effect of isolating each type of information. We name this method SciBERT-fake-token.

We also attempt to add words like "[method]", "[abstract]", "[sid=1]" as special tokens into the dictionary of SciBERT so that the tokenizer would not split them into pieces. We call this method as SciBERT-special-token. The results on the development set of all the strategies based on BERT framework are given in Table 3. The results on the blind test set are shown in Table 7 and Table 8.

SemBERT-based method:
The models based on the BERT framework are not able to leverage other information besides contextualized semantics knowledge. We propose the method based on Semantics-aware BERT for Language Understanding (SemBERT) **(Zhang, Zhuosheng et al., 2020).** The existing language representation models including ELMo, GPT, and BERT only exploit plain context-sensitive features such as character or word embedding. They rarely consider incorporating structured semantic information which can provide rich semantics for language representation. To promote natural language understanding, Zhang, Zhuosheng et al. proposes to incorporate explicit contextual semantics from pre-trained semantic role labeling and introduces an improved language representation model.
In this work, we use the pre-trained semantic role labeling model, trained on The Stanford Natural Language Inference (SNLI) Corpus, to offline annotate the train and development set. The annotation is an unsupervised process, which makes the annotation task more lightly. Instead of BERT, the language model part is replaced as SciBERT.
With the structured semantic information extracted, we use the sentence pairs and the structured semantic information of every sentence as the input of SemBERT and then select the top-3 candidates as positive pairs. The performance is reported in Table 4, Table 9, and Table 10.

BERT-independent classifier methods:
How to utilize document-level features is still worth discussing. We apply two-stage training methods to combine information from different levels. There are three types of features to be considered in our proposed systems in this part:
Sentence-pair Embeddings. We keep the weights of best fine-tuned SciBERT and ACLBERT models and leverage them as a text encoder in this section. The embedding of token "[CLS]" of the last layer of BERT part is taken as the sentence-pair embedding. Therefore, a vector with the length of 768 or 1024 is gained for a sentence-pair, from SciBERT and ACLBERT, respectively.

Section Embeddings. First of all, we train classification with center loss for nine labels (section categories mentioned in the BERT-based methods part), then take the nine center embedding as fixed embedding for each sections (Qi, Ce et al., 2017). The embedding dimension is set to 32 in our experiments. To illustrate, we also use the 1000 documents in the training process.

Position Features. Three features of position are generated: "sid", "ssid" are directly used as integers. Besides, the value of "sid" over the length of the reference paper is also calculated as the relative position.

We do experiments with different input choices on four classification models without neural network: Random Forest (RF), Logistic Regression (LR), CatBoost (Dorogush, Anna et al., 2018) and LGBM. The systems are shown in Table 5. SciBERT means using all three types of features and the sentence pair embeddings are generated from fine-tuned SciBERT. Similarly, ACLBERT utilizes the sentence pair embeddings from fine-tuned ACLBERT. The sentence pair embeddings used in SciBERT-ACLBERT are the concatenation of the embeddings from two SciBERT and ACLBERT.

The sentence pair embeddings from SciBERT or ACLBERT may weaken the impact of section embeddings and position features due to the high dimension, so that we try to use the scores of the prediction to replace the token embedding. We concatenate the prediction score of the best 4 models (SciBERT and ACLBERT) with section and position features as the input of classifiers. This System is named Four-output in table 5.

The scores on the blind test set are given in Table 11 and Table 12.

3.2 TASK 1B

Task 1B is a task of multi-label classification. The five facet labels are "Implication", "Hypothesis", "Aim", "Results", "Method". In this part, only all cited text spans, and their position and section information are taken into account. We train a multi-label model with the initial weights of SciBERT. To deal with the imbalance of data, we reset the class weights negatively correlated with the sample size for binary cross-entropy loss. And the thresholds of "Implication""Hypothesis" are set less than 0.5 to improve the recall scores. The results are shown in Table 6.

4 Result and Discussion

4.1 TASK 1A

To evaluate the methods, we regard the Sentence Overlap F1 scores of development set as the performance of our proposed methods (Chandrasekaran, Muthu et al., 2020). We sort prediction scores of all candidate sentences for each citing sentence and then keep the top 2 or 3 as the final selection.

The precision, recall, and F1 scores are calculated for every model based on the BERT framework in Table 3. It is noted that the results in table 3 are the assembled results of two best models, so it is likely overfitted on the development set, but these values also make sense to compare the performance of different methods.

Table 3 illustrates that ACLBERT possesses better performance than SciBERT. The use of fake token does make the performance improved, whereas the supplements of special tokens make the F1 score decrease. This is also reasonable because there is not enough data to train tokens that did not appear in pre-training.

Method	TopN	Precision	Recall	F1
SciBERT	N=2	0.1940	0.2949	0.2342
SciBERT-special-token	N=2	0.1606	**0.3653**	0.2231
SciBERT-fake-token	N=2	**0.2038**	0.3077	**0.2452**
ACLBERT	N=2	0.2029	0.3076	0.2446

Table 3: Results of BERT-based methods

Method	TopN	Precision	Recall	F1
SciBERT	N=3	0.1940	0.2949	0.2342

Table 4: Results of SemBERT

Input	RF	LR	CB	LGBM
SciBERT	*0.27*	0.201	0.1959	0.1946
ACLBERT	*0.2392*	*0.2467*	0.2222	0.2239
SciBERT - ACLBERT	*0.2435*	0.2141	*0.2386*	*0.2545*
Four-output	*0.2615*	*0.2612*	0.2357	0.2155

Table 5: Results of BERT-independent classifier

Input	Precision	Recall	F1
Aim	1.0	0.4285	0.6
Hypothesis	0.0727	1.0	0.1355
Implication	0.2603	1.0	0.4131
Method	0.8223	0.9842	0.8960
Results	0.4655	0.9310	0.6206

Table 6: Results of TASK 1B

SciBERT-fake-token is the best strategies based on BERT framework, followed by ACLBERT.

Table 4 shows the result of SemBERT. This result is fairly reliably with minimal over-fitted effect, though the value is lower than those in Table 3. It is noticed that SemBERT gains a considerable recall score, compared to the methods in Table 3.

Table 5 shows the results of the BERT-independent classifier methods. The scores cannot be compared with those in Table 3 and Table 4 because we are based on the most excellent BERT-based models. We choose the systems marked in Italic as the final result for systems based on the BERT-independent classifier framework to submit.

4.2 TASK 1B

We train repeatedly SciBERT models for with different random seeds, then choose the best model for each label, as Table 6. It has to be mentioned that the results of "Aim" and "Hypothesis" are not stable under different random seeds without sufficient training data. "Method" category has the highest score, while "Hypothesis" gains the lowest one.

Method	TopN	Precision	Recall	F1
SciBERT	N=2	0.1178	0.2182	0.1530
SciBERT (5cv)	N=3	0.1043	**0.2901**	0.1534
SciBERT-special-token	N=2	0.1183	0.2224	0.1545
SciBERT-fake-token	N=2	0.1189	0.2238	0.1552
SciBERT-fake-token(5cv)	N=2	**0.1292**	0.2417	**0.1684**
ACLBERT	N=2	0.1143	0.1989	0.1451

Table 7: Sentence Overlap scores of BERT-based methods

Method	TopN	Precision	Recall	F1
SciBERT	N=2	0.2664	0.1063	0.1352
SciBERT (5cv)	N=2	0.2757	0.1044	0.1361
SciBERT-special-token	N=2	0.2820	0.1035	0.1394
SciBERT-fake-token	N=2	0.2741	0.0892	0.1202
SciBERT-fake-token(5cv)	N=2	**0.2950**	**0.1174**	**0.1498**
ACLBERT	N=2	0.2620	0.0924	0.1240

Table 8: ROUGE-SU4 scores of BERT-based methods

Method	TopN	Precision	Recall	F1
SciBERT	N=2	0.1167	0.2155	0.1515

Table 9: Sentence Overlap scores of SemBERT

Method	TopN	Precision	Recall	F1
SciBERT	N=2	0.2634	0.0903	0.1232

Table 10: ROUGE-SU4 scores of SemBERT

5 Submitted Runs

For Task 1A we submit the results of 27 strategies, where each strategy contains two results (top 2 candidates, or top 3 candidates). Among all the submitted runs, the ensemble of SciBERT, SciBERT-fake-token, SciBERT-special-token, and SemBERT based on SciBERT, named "SciBERT_SemBERT" in

Input	RF	LR	CB	LGBM
SciBERT	0.1161	0.1253	0.1303	0.1316
ACLBERT	0.1155	0.1241	0.1335	0.1349
SciBERT-ACLBERT	0.1095	0.1293	**0.1366**	**0.1408**
Four-output	**0.1260**	**0.1440**	/	/

Table 11: Sentence Overlap scores of BERT-independent classifier methods

Input	RF	LR	CB	LGBM
SciBERT	0.0970	0.1097	0.1056	0.1066
ACLBERT	0.0911	0.1109	0.1096	**0.1179**
SciBERT-ACLBERT	0.0922	0.1120	**0.1210**	0.1118
Four-output	**0.1105**	**0.1225**	/	/

Table 12: ROUGE-SU4 scores of BERT-independent classifier methods

Method	TopN	Precision	Recall	F1
SciBERT-ALL	N=3	0.1139	0.3178	0.1677
SciBERT-SemBERT	N=2	0.1318	0.2459	0.1716
SciBer-ACLBERT	N=2	0.1244	0.2265	0.1606

Table 13: Sentence Overlap scores of mixed strategies

Method	TopN	Precision	Recall	F1
SciBERT-ALL	N=2	0.2860	0.1105	0.1433
SciBERT-SemBERT	N=2	0.2976	0.1134	0.1470
SciBer-ACLBERT	N=2	0.2905	0.1029	0.1387

Table 14: ROUGE-SU4 scores of mixed

TASK	Precision	Recall	F1
Task1B	0.1914	0.2899	0.2306

Table 15: Sentence Overlap scores of Task1b

Table 13 and Table 14, wins the highest Sentence Overlap F1 scores (Micro F1: 0.1716, Macro F1: 0.1737). In the meanwhile, the ensemble of SciBERT-fake-token models trained for 5-fold cross-validation gains the highest ROUGE-SU4 F1 score (0.1498), shown in Table 8.

The scores of strategies based on the BERT Framework have been shown in Table 7 and Table 8. Among three single models, SciBERT-fake-token shows the best performance (Sentence Overlap Micro F1: 0.1552), which is consistent with the scores on the development set. SciBERT-special-token gets the highest ROUGE-SU4 F1 score (0.1394). SciBERT-special-token and SciBERT-fake-token perform better, which indicates that the position and section information has a positive effect. Unlike results on the development set, ACLBERT does not perform worse than SciBERT. We submitted the results of two strategies (SciBERT and SciBERT-fake-token) trained for 5-fold cross-validation, with Sentence Overlap Micro F1 0.1534 and 0.1684 respectively. The scores show that the ensemble of models for 5-fold cross-validation outperforms the corresponding individual models.

Table 9 and Table 10 illustrate the performance of SemBERT based on SciBERT of which the Sentence Overlap F1 score is 0.1515 and ROUGE-SU4 F1 score is 0.1232.

The scores of the BERT-independent classifier methods are given in Table 11 and Table 12. The scores are lower than BERT-Based methods and SemBERT-based methods. The figures demonstrate that the end-to-end trained models based on BERT outperform models trained with two independent stages.

We also fuse various strategies and the results are shown in Table 13 and Table 14. Overall, the performance proves better than a single strategy, with all Sentence Overlap F1 scores more than 0.16.

We can be convinced that the ensemble of different models lessens the impact of over fitting, although we only trained on the 40 manually annotated articles.

The evaluation of Task1B is given in Table 15.

6 Conclusions and Future Work

In our systems proposed, the semantic knowledge inside citing sentences and their corresponding cited text spans is encoded by models based on the BERT Framework. Besides, we do the experiments on how to integrate the structured semantic information as well as document-level information. In the result section, we analysis the performance and choose good system of all the proposed models as final systems to submit. The scores on the blind test set indicates position and section information can improve the identification of cited text spans. The end-to-end trained models outperform models trained with two stages, and the averaged prediction of multi-models is more accurate than an individual one. How to learn more extra information should be further explored. The relation between sentences in the same paper is not considered in our methods, which is one of the points we can do in the future work.

References

Abu-Jbara, Amjad & Radev, Dragomir. (2011). Coherent Citation-Based Summarization of Scientific Papers. 500-509.

Chandrasekaran, Muthu & Yasunaga, Michihiro & Radev, Dragomir & Freitag, Dayne & Kan, Min-Yen. (2019). Overview and Results: CL-SciSumm Shared Task 2019.

Chandrasekaran, M. K., Feigenblat, G., Hovy. E., Ravichander, A., Shmueli-Scheuer, M., De Waard, A. (Forthcoming). Overview and Insights from Scientific Document Summarization Shared Tasks 2020: CL-SciSumm, LaySumm and LongSumm. In Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020).

Nomoto, T. (2018). Resolving citation links with neural networks. Frontiers in Research Metrics and Analytics, 3, 31.

Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Bird, Steven, Edward Loper and Ewan Klein (2009). Natural Language Processing with Python. O'Reilly Media Inc.

Beltagy, Iz & Lo, Kyle & Cohan, Arman. (2019). SciBERT: A Pretrained Language Model for Scientific Text. 3606-3611. 10.18653/v1/D19-1371.

Zhang, Zhuosheng & Wu, Yuwei & Zhao, Hai & Li, Zuchao & Zhang, Shuailiang & Zhou, Xi & Zhou, Xiang. (2020). Semantics-Aware BERT for Language Understanding. Proceedings of the AAAI Conference on Artificial Intelligence. 34. 9628-9635. 10.1609/aaai.v34i05.6510.

Qi, Ce & Su, Fei. (2017). Contrastive-center loss for deep neural networks. 2851-2855. 10.1109/ICIP.2017.8296803.

Dorogush, Anna & Ershov, Vasily & Gulin, Andrey. (2018). CatBoost: gradient boosting with categorical features support.

Wang, Zhiguo & Hamza, Wael & Florian, Radu. (2017). Bilateral Multi-Perspective Matching for Natural Language Sentences. 4144-4150. 10.24963/ijcai.2017/579.