NaCTeM-UoM @ CL-SciSumm 2019

Chrysoula Zerva, Minh-Quoc Nghiem,
Nhung T.H. Nguyen, and Sophia Ananiadou

National Centre for Text Mining, University of Manchester, UK
{chrysoula.zerva, minh-quoc.nghiem}@manchester.ac.uk
{nhung.nguyen, sophia.ananiadou}@manchester.ac.uk

Abstract. This paper introduces the National Centre for Text Mining - University of Manchester systems submitted in CL-SciSumm 2019 Shared Task at BIRNDL 2019 Workshop. CL-SciSumm shared tasks focus on the identification of cited passages across scientific publications, and the subsequent summarisation of scientific articles based on their cited extracts. More specifically Task 1A is directed at the identification of cited text spans in the reference paper, based on the provided citation passages, while Task 1B concerns the classification of the citation passages based on their function in the text. For Task 2, the identified cited text spans are used in order to generate an informed summary for the reference paper. We participated in both tasks described above. We looked into supervised and semi-supervised approaches and explored the potential of adapting bidirectional transformers for each task. We further formalised Task 1A as a similarity ranking problem and implemented bilateral multi-perspective matching for natural language sentences.

Keywords: citation extraction · scientific summarisation · BiMPM · BERT · sentence similarity

1 Introduction

In scientific publications, citations can serve a range of different functions, targeting different aspects of the referenced publication. For example, some citations aim to compare to methods or results of the referenced paper, some intend to build upon the cited methods, while others aim to corroborate or dispute a given hypothesis or conclusion. [22]. Hence, different citations to the same paper may refer to different text spans within that paper. Nevertheless, citations are expected to focus on the most important and mention-worthy aspects of a publication. Thus, the combination of the citations referring to the same publication is believed to be indicative of its main points and contributions [18].

The aforementioned observations kindled the interest in citation-based summarisation methods for scientific publications, which aim to combine information from citing sentences in order to improve the summary of the referenced article [5, 1]. However, citing sentences are expected to include the opinion of the citing author(s) alongside the information about the referenced publication, and disentangling between the two can prove to be a particularly complicated task.

For this reason, it has been proposed that cited text spans of the referenced article could provide less biased information to support the scientific summarisation task. The CL-SciSumm Shared Tasks [10, 8, 9, 4] are built around this idea, proposing a set of sub-tasks that address the different steps that could lead to a more efficient scientific summarisation system, informed by cited text spans.

More specifically, the CL-SciSumm 2019 task [4] is formulated as follows: Given a set of reference papers (RP) and their corresponding papers that cite them (CP), participants have to build systems that can address Tasks 1A, 1B and (optionally) Task 2.

T1A: For each citance (i.e. a citation sentence that references the RP), identify the spans of text (cited text spans) in the RP that most accurately reflect the citance.

T1B: For each cited text span, identify what facet of the paper it belongs to, from a predefined set of facets namely: Method, Aim, Implication, Results or Hypothesis.

T2: (Optional) Generate a structured (of up to 250 words) summary of the RP.

We approached Task 1A in two different ways, using sentence similarity and sentence pair classification methods. In the first approach, we investigate the potential of the BERT bidirectional transformer model [6], when applied to the classification of citing-cited sentence pairs. BERT has shown great potential in sentence-pair classification tasks while BERT-based embeddings have been demonstrated to efficiently capture context in a wide range of different tasks [6, 11, 21, 26] . In the second approach, we employed bilateral multi-perspective matching model [23] to calculate the similarity between RP and CP sentences. The model has successfully applied to three tasks: paraphrase identification, natural language inference and answer sentence selection. For Task 1B, due to the small dataset size, we experimented with a RF classifier alongside a BERT-based approach.

Looking at the summarisation (Task 2), there are generally two approaches currently being adopted to create a summary from an input text: extractive and abstractive. Extractive summarisation produces a summary by choosing a subset of sentences related to the main idea of the input document. Abstractive summarisation, in contrast, generates summaries by modifying phrases and sentences from the input. It is relatively difficult to properly create attractive summaries because it requires semantic analysis. To ensure we get grammatically correct summary for Task 2, we focus on creating the summary by using extractive methods.

For this task, the system needs to generate a structured summary from the cited text spans of the input reference paper. While cited text spans capture the main points of interest for the scientific community, the paper's full text gives more detailed information about its content, which is useful for the summary. According to our analysis on the provided training set, only a small amount of text (16%) was taken from the cited text spans while a majority of them (76%) was from the rest of the full text. Because of this, besides using cited text spans, we also employ the full text of the paper in our approach.

2 Data Pre-processing

2.1 Task 1

For Task 1, the organisers provide two different datasets for training: (1) a manually annotated dataset comprising 40 articles and their respective citing papers, which was also used in the 2018 CL-SciSumm challenge, and (2) a dataset of 1000 articles and their respective citing papers, which were automatically annotated with a neural network approach as described in [15]. Henceforth, we will refer to the first dataset as the *2018 dataset*, the second one as the *2019 dataset* and their combination as the *2018-2019 dataset*. Of those only the 2018 dataset contained annotations for Task 1B, and was used for the related experiments.

In order to estimate the performance of the methods discussed in Sections 3.1 and 3.2, we keep 20% of the 2018 dataset (eight randomly selected articles) on the side to be used as a development set[1].

For Task 1A it was necessary to extract sentence pairs between citing sentences and text spans from the Reference Papers (RP), that would then be used as training instances for our models to learn how to classify or rank such pairs as valid or invalid citing-cited sentence pairs. We pre-processed the provided annotation files (.ann) as well as the XML files for the RP in order to extract positive and negative pairs. For the positive pairs, we used the sentences as provided in the *.ann* files. We applied a set of sentence reconstruction rules to sentences that were erroneously segmented by the OCR (e.g., erroneously segmented after parentheses, enumeration or abbreviations)[2]. The same pre-processing was applied to all sentences of the RP.

For the generation of negative pairs, each citation sentence was paired to randomly selected sentences from the RP. The RP sentences to be used for the negative pair generation were further processed as following: Each candidate sentence was tokenised[3], and then each token was lemmatised. Subsequently each lemma was looked up against WordNet [14] and a combination of stopword lexica to estimate whether it is a valid word or an OCR error. If $< 50\%$ of the candidate sentence lemmas is found to be a valid word, the sentence is rejected. Apart from this filtering step, no further processing to alter the OCR output was applied. In order to keep a balance between adequate training data and label imbalance, we chose a proportion of 4 negative pairs per citance. The same processing is applied on both the 2018 and 2019 dataset.

Overlap-controlled pair generation. As shown in Table 1, between a CP and RP sentences there is a certain percentage of overlapping vocabulary[4]. Nearly

[1] The ids of the papers used for validation are: C00-2123, C04-1089, I05-5011, J96-3004, N06-2049, P05-1004, P05-1053, P98-1046

[2] The original *sid* and *ssid* offsets of the reconstructed sentences were indexed and restored for the final system outputs.

[3] NLTK Tokenizer was used for the tokenisation in all pre-processing steps

[4] It is noted that we removed stop words and symbols when calculating overlapping vocabularies.

half of the positive pairs in the 2018 training set have a vocabulary overlap of size >= 2. In an attempt to assess and control the impact of word overlap on the information learned by our models, we also experimented with the henceforth called "overlap-controlled" generation of negative pairs. In this case, negative pairs were selected so that the word overlap between the citing sentence and the reference sentence was maximised. Hence we obtained two additional datasets, the 2018 overlap-controlled (OV) dataset and the 2018-2019 overlap-controlled (OV) dataset.

Table 1. Statistics of vocabulary overlap in positive and negative pairs between RP and CP sentences in the 2018 dataset.

Num. overlap vocab.	Train-pos.	Train-neg-rand.	Train-neg-OV.	Dev-pos
0	315	1,252	2,594	79
1	306	898	2,000	74
2	229	295	638	65
3	145	101	269	64
4	77	47	104	38
5	48	12	43	13
6	26	8	19	6
7	11	1	10	2
8	7	1	8	4
9	4	0	4	1
>=10	9	1	5	1

The position of the RP sentence within the document (*sid* offset) and the document section (*ssid* offset) were also encoded in the generated pairs, and used as additional feature in some of the Task 1A methods (see feature-based BERT approach) as well as for Task 1B.

2.2 Task 2

In order to prepare the data for Task 2, we first filter out too long (more than 45 tokens) or too short (less than 5 tokens) sentences. Any unrelated sentences (i.e., sentences which belong to "Acknowledgment" or "References" sections) are also removed. We then tokenise the text using the stanford-corenlp toolkit[5].

The provided training data (2018 dataset) was created using abstractive summarisation methods, which are not suitable to use for extractive summarisation models. To identify which sentences should be put into the extractive summary, we greedily selected sentences which can maximise the ROUGE scores to create an extractive summary version of the originally provided data. To generate training data for the classifier (described in Section 3.3), we assigned label 1 to sentences selected in the extractive summary version and 0 otherwise, thus obtaining positive and negative instances.

[5] https://stanfordnlp.github.io/CoreNLP/

3 Methods

3.1 Task 1A

We considered two main approaches for the identification of cited text spans, both centred around the concept of identifying sentence relevance/similarity between the citing and cited text spans.

BERT-based model. In the first approach we explore the potential of fine-tuning bidirectional transformers, and more specifically the pre-trained BERT model [6]. Through unsupervised pre-training of language models on large corpora BERT has been shown to significantly improve the performance on many NLP tasks, including tasks which aim to identify relevance between two text spans (e.g. SQUAD [20]). It was thus deemed suitable to experiment with for Task 1A. Moreover, BERT is pre-trained on a *language modelling* (LM) and a *next sentence prediction task*. Hence, BERT's architecture and learned embeddings account for sequence pairs and can be adapted for Task 1A.

For all experiments that use BERT models, the relevant code is implemented in python and using the pytorch library. The BERT-based classifiers are built on top of BERT models as provided in pytorch by huggingface on github [6].

We used the *bert-base-uncased model* for the experiments, which has the following set-up: 12 layers, hidden vectors of size 768 and 12 self-attention heads. We initially fine-tuned the model trained for the next sequence classification task on both the 2018 dataset and the 2018-2019 dataset, as well as the respective OV versions. For the 2018-2019 dataset we used two training approaches:

1. Use the 2018-2019 dataset and randomly sample batches for all epochs.
2. Start with (1) until convergence and then continue by sampling only from the 2018 dataset for a few epochs, using weight and learning rate decay (henceforth referred to as 2018FT approach).

We also experimented with using BERT base model in a feature-based approach, to extract features that were then used as input in a Convolutional Neural Network (CNN). For this purpose we used the concatenation of the last 4 layers of the BERT model to extract a feature vector for each token as it has been shown to achieve optimal performance according to [6]. The CNN used for the experiments consists of three convolution layers, followed by a fully connected linear layer. We use 3x3 AvgPooling after each convolution layer and a dropout of 0.1 after the last convolution layer.

For the feature-based approach, the position of the RP sentence in the document (*sid* offset) and the position of the RP sentence in the section (*ssid* offset) were also added as features. We call those features position features and they are concatenated with the CNN output and used as input for the linear layer.

[6] https://github.com/huggingface/pytorch-pretrained-BERT version 0.4.0, which has been verified to reproduce the outputs of the original TensorFlow implementation.

Since BERT is pre-trained on data from the general domain, we wanted to also experiment with models closer to the CL-SciSumm domain. For this reason we employed two different approaches:

1. Fine-tuning the weights of the bace model on the ACL anthology reference corpus (ACL-ARC) [19] and then train on the CL-SciSumm data as above
2. Use the SciBERT model [3] that is pre-trained on a collection of 1.14M documents from Semantic Scholar [2].

For fine-tuning on the ACL-ARC corpus, we aimed to fine-tune the BERT base model weights for the next sentence prediction task. We pre-process the corpus to filter out sentences with low OCR quality. For the sentence filtering we first use the OCR parsing confidence score, and reject sentences with score $<= 0.6$. Subsequently we use a rule-based approach to correct sentences that have been erroneously segmented by the OCR. We then end up with a set of 7M sentence pairs. Of those, half are consecutive sentences and the rest randomly chosen sentence pairs. We use the fine-tuning approach described in [7] and fine-tune for 3 epochs and a batch size of 16, with initial learning rate, $LR = 3E-5$. We refer to this fine-tuned version as the *ACL model*. We then repeat the experiments that were performed with the BERT base model; the performance can be observed in Table 4. The SciBERT model was used with the feature-based approach described for the BERT base model, without further fine-tuning. We provide the performance results in Table 5.

Bilateral multi-perspective matching model. With the intuition that there is probably some correlation between citing and cited text spans, e.g., they may be paraphrase of each other or they may have some inference relation, we employed Bilateral Multi-Perspective Matching model (BiMPM) [23] for Task 1A. BiMPM firstly encodes two input sentences with BiLSTM and then matches the encoded ones in both directions (from left to right and from right to left). In the matching stage, the model uses four matching strategies to compare each time-step in one sentence against all time-steps in the other sentence.

In this work, we used Glove pre-calculated embeddings [17] as input to BiMPM. We considered each pair of citing and cited sentences as a positive pair while generating negative pairs by the two aforementioned ways. As a result, we conducted four experiments (for the 2018, 2019 datasets and OV versions) with 100 epochs and a batch size of 6. In the testing stage, we firstly calculated scores of pairs between citing texts and all CP sentences and then selected the top-3 candidates as positive pairs. The performance is reported in Table 6.

3.2 Task 1B

For Task 1B we did not use any information from the citing sentence. The features were generated exclusively based on the identified cited text of the RP. While it is a multi-class, multi-label problem we concluded in building separate binary classifiers for each facet label and subsequently concatenating the positive

output for each label. The motivation behind this approach is the imbalance in the label proportions of Task 1B (see Figure 1).

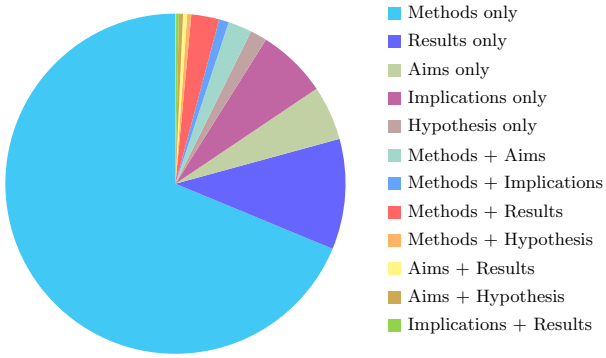

Fig. 1. Proportion of Task 1B labels and their combination in the 2018 dataset.

We experimented with two approaches: (1) A Random Forest (RF) classifier [12] and (2) a BERT classifier using an adaptation of the BERT for binary single-sentence classification tasks as described in [6]. For both approaches we use token-based features and sentence-position features (*sid* and *ssid* offsets). The initial feature extraction steps that converted the training instances (RP cited text spans) to features were the same in both approaches. The BERT Tokeniser (WordPiece tokeniser [24]) for tokenisation and a limit of max 512 tokens per instance was imposed.

For the RF classifier implementation we used the scikit-learn [16]. We used vectorised token representations removing stop-words and using a BOW approach for vectorisation. We then concatenated the token vectors with the position features. We trained a RF classifier with 1000 trees for each facet.

For the BERT classifiers, we used the same set-up described in Section 3.1 for the feature based approach. We use the BERT sentence classification model provided for pytorch by *huggingface*[7], and we train separate binary classifiers for each facet type.

Both approaches proved to be weak in identifying "*Hypothesis*" cases, probably because of the very low amount of training data (only 18 instances). For that reason we used an adaptation of the rule-based approach described in [25] to identify hypothetical and investigative sentences which we annotate as "*Hypothesis*". We added the rule-based approach as an additional rule-based classifier. Thus the final output for Task 1B is formulated as the union of the positive outputs of the individual facet classifiers. If all classifiers return 0, we return the "*Methods*" label as a default.

[7] https://github.com/huggingface/pytorch-pretrained-BERT version 0.4.0

3.3 Task 2

We formulate the summarisation task as a classification problem. The classifier needs to classify the sentences in the input document into two classes: included or not included in the summary. We then rank the sentences based on how likely they are to be included in the final summary. From the ranked list, we add the sentences into the final summary one by one ensuring that there is no trigram overlap between the current summary and the sentence to be added. The process stops when the summary reaches the maximum length (250 words in this task).

The classifier we use is similar to the one of Liu [13]. We employ the sentence vectors from BERT but using multiple `[CLS]` symbols to get features for multiple sentences. Odd sentences are assigned a segment embedding E_A while even sentences are assigned a segment embedding E_B. Finally, a linear model is added to BERT output to predict the score for each sentence (1 is included, 0 is not included).

The small size of the CL-SciSumm dataset rendered it harder to train any neural model. To solve this, we train all of our models using the data from SciSummNet. The benefit of this approach is that we can take advantage of its large size. The drawback, however, is that all summary sentences in SciSummNet were taken from the original paper which makes them all subjective sentences. After we obtain the summary, we apply simple rule-based heuristics (for example, change "our" to "their") to convert the subjective sentence to an objective one.

4 Results and Discussion

4.1 Task 1A

For the evaluation results presented in this section, we generate all possible citing-cited sentence pairs for each RP and then apply the trained models on each pair. For each citing text span we choose the top three scoring pairs and return them as the predicted positives [8].

Our observations on the training set show that RP sentences are repeatedly cited from different CP citing sentences. Table 2 shows that half of the RP sentences are cited twice while the others are cited from 3 to 17 times in the 2018 dataset. We observed that this fact might be biasing our models in favouring specific sentences, but it is also significantly affecting the calculated performance in the case of missing highly repeated sentences.

BERT-based model. The experiments on directly fine-tuning the pre-trained BERT base model are presented in Table 3. We notice that the best performance is obtained when fine-tuning exclusively on the 2018 dataset, despite the small number of training samples. When incorporating the automatically annotated

[8] The BERT-based classifiers output a score for each class [invalid, valid]. The top three scores for the "valid" annotated pairs are used. If there is no "valid" pair for a given citing text span we return the best scoring "invalid" pair

Table 2. The number of RP sentences that are repeatedly cited.

dataset	2 times	more than 2 times
2018-train	122	132
2018-dev	40	38
2019	2,996	2,001

2019 data, the false positive rate increases for all classifiers. Even when using the 2018 dataset exclusively at the end of the training (2018FT), the classifier cannot exceed performance obtained on 2018 dataset. Moreover, the approach of using the overlap-controlled datasets for training yields lower performance results for all models. This could be attributed to the fact that by controlling the word overlap between candidate sentence pairs, we are implicitly reducing the vocabulary size that we fine-tune on, leading to classifiers that do not generalise well on unseen data.

In Table 4 we can see the results for the ACL model. Based on the results of previous experiments (see Table 3) we refrained from evaluating the performance on the overlap-controlled datasets. We can see that we obtain a small improvement, both on the 2018 and 2018-2019 datasets. The addition of the 2018-FT however, does not boost performance as in the case of the BERT base model. Still, we can argue that fine-tuning the pre-trained model on data from the specific target domain can aid in improving the model.

Finally we present the results from the feature-based BERT experiments, using the BERT-base and the SciBERT models. We evaluated those models only on the 2018 dataset, as shown on Table 5. Both models reach similar performance, without a significant boost from the SciBERT approach. This could be attributed to the higher proportion of biomedical documents compared to computer science ones, in the training data used for SciBERT. Hence the model might be a better fit for the biomedical domain.

Table 3. Performance for the BERT base model fine-tuning on the CL-SciSumm task.

dataset	Recall	Precision	F1-score
2018	**0.325**	0.161	**0.215**
2018-2019	0.277	0.144	0.189
2018-2019 + 2018FT	0.295	**0.167**	0.205
2018 OV	0.113	0.103	0.108
2018-2019 OV	0.094	0.133	0.110
2018-2019 OV + 2018FT	0.227	0.156	0.185

BiMPM model Table 6 reports the performance of BiMPM model on the development set. Although the 2019 dataset was automatically generated, by combining it with the golden 2018 dataset, we could obtain the best performance,

Table 4. Performance results for the ACL model trained on CL-SciSumm data.

dataset	Recall	Precision	F1-score
2018	**0.334**	**0.171**	**0.226**
2018-2019	0.275	0.155	0.198
2018-2019 + 2018FT	0.278	0.161	0.204

Table 5. Performance results for the feature based approach.

feature embeddings	Recall	Precision	F1-score
BERT	**0.141**	0.243	0.178
SciBERT	0.135	**0.264**	**0.179**

which was significantly better than that on the 2018 dataset. Meanwhile, using the overlap-controlled strategy for generating negative pairs could slightly improve the performance on the 2018 dataset but not on the 2018-2019 dataset.

Table 6. Performance of the BiMPM model on the development set when top-3 candidates were selected as positive pairs.

dataset	Recall	Precision	F1-score
2018	0.016	0.007	0.010
2018 OV	0.019	0.008	0.012
2018-2019	**0.113**	**0.046**	**0.066**
2018-2019 OV	0.042	0.067	0.052

Similarly to the above-mentioned situation in the training set that RP sentences were repeatedly cited in reference papers (see Table 2), the BiMPM model also favoured some certain RP sentences. For example, with the 2018 dataset, the model could predict only 61 sentences as cited text spans for 349 CP sentences of the development set, which explains why its performance was unexpectedly low. In the case of the 2018-2019 dataset, the number of predicted RP sentences was 151, which is more diverse than that of the 2018 one.

4.2 Task 1B

In Table 7 we can see that the BERT classifier obtains better performance for the highly represented labels, but fails to learn the underrepresented ones. The RF classifier seems to perform better on those labels, but it should be noted that when applied on the testing data it failed to identify any "*Hypothesis*" citations.

4.3 Task 2

We use ROUGE-1, ROUGE-2, and ROUGE-L scores for evaluation. We use ScisummNet data for training and report the result on the CL-Scisumm training

Table 7. Performance of the RF and BERT citation facet classifiers

	RF classifier			BERT Classifier		
	Precision	Recall	F-score	Precision	Recall	F-score
Methods	0.86	**0.92**	0.89	**0.89**	**0.92**	**0.90**
Aims	**0.67**	**0.55**	**0.60**	0.27	0.12	0.17
Implication	**0.33**	**0.11**	**0.17**	0.00	0.00	0.00
Results	**0.68**	0.49	0.57	0.65	**0.55**	**0.60**
Hypothesis	**0.13**	**0.45**	**0.20**	0.00	0.00	0.00

data 2019 as well as the CL-Scisumm test data 2016. All models use BERT base uncased model with 50,000 training steps. Table 8 show the results on four different settings where the model selects the sentences from.

Table 8. Performance on CL-SciSumm data

	ROUGE-1	ROUGE-2	ROUGE-L
CL-Scisumm training data 2019			
Select sentences from all text	47.73	22.05	45.35
Select sentences from abstract + community	49.29	24.32	46.57
Augment abstract + all text	47.30	21.76	44.99
Augment abstract + community	**49.36**	**24.66**	**46.70**
CL-SciSumm test data 2016			
Select sentences from all text	48.98	24.63	46.68
Select sentences from abstract + community	46.03	23.67	43.59
Augment abstract + all text	**49.52**	**25.44**	**47.20**
Augment abstract + community	46.19	23.74	43.76

Based on our observations, most of the summary sentences are selected from the beginning of the input document. Indeed, the abstract alone can yield the best ROUGE-2 score (25.54), although the ROUGE-1 and ROUGE-L scores are lower than the scores in our proposed method. This result may be explained by the fact that the abstract has already conveyed most of the ideas described in the paper. It is also because of the way the training data (ScisummNet) was created: the human annotators only read the abstract and the cited text spans from the paper.

5 Submitted Runs

For Task 1A we submitted 11 runs, and used the RF classifier for Task 1B. We can see that similarly to our experiments in Section 4.1 the ACL model seems to outperform other approaches. However, with the exception of the BiMPM model (run 11), most systems show a significant drop of performance when applied on the testing data, pointing to weak generalisation of the models. Still, the ACL model outperformed other submissions in the 2019 CL-SciSumm task.

Table 9. Submitted system and obtained performance for each run in Task 1

Run	System	1A: Sent. Ov. (F1)	1A: R-SU4 (F1)	1B (F1)
1	BERT 2018	0.093	0.06	0.255
2	ACL 2018	**0.126**	0.075	**0.312**
3	BERT 2018/19	0.097	0.062	0.251
4	BERT 2018/19+2018FT	0.11	0.062	0.283
5	BERT 2018/19 OV+2018FT	0.12	0.072	0.303
6	ACL 2018/19 + 2018FT	0.118	**0.079**	0.292
7	SciBERT 2018	0.078	0.048	0.218
8	BiMPM 2018 OV	0.074	0.051	0.221
9	BiMPM 2018/19	0.012	0.018	0.039
10	BiMPM 2018 OV top2	0.11	0.073	0.276
11	BERT 2018 top2	0.062	0.052	0.15

Table 10. Submitted system and obtained performance in Task 2

	2: R-2 (F1)	2: R-SU4 (F1)
Abstract	**0.514**	**0.295**
Community	0.106	0.062
Human	0.265	0.180

For Task 2, we submitted only one model which augments the original abstract of the paper using sentences from the full papers to create the summary. Table 10 shows the results obtained from the submitted system on the testing data. The best score is obtained with the abstract-based evaluation, which can be explained since we opted for an abstract augmenting approach.

6 Conclusions

We have described the systems developed to participate in Tasks 1A, 1B and 2 in the CL-SciSumm 2019 shared task. For Task 1A we implemented two methods, which use neural networks to learn the relation between citing and cited text spans; bidirectional transformers (BERT-based) and BiLSTM networks (BiMPM-based). We showed that the BERT-based models could efficiently be trained on the manually annotated data, but could not benefit from automatically annotated one. Instead the BiMPM-based method showed significant improvement when trained on large data, even if it was automatically annotated (i.e., noisy). For Task 1B, we resorted to using an RF classifier over a BERT-based approach, since it could handle a smaller training dataset and under-represented labels better.

On Task 2, in order to take advantage of the informative sentences that authors provided in the abstracts, we augmented the abstract with selected sentences from the full text. The experimental results have shown that this approach outperformed the one that only used extracted sentences from full text.

Acknowledgments. This work was partly supported by the EPSRC Doctoral Prize award; the HSE Discovering Safety, Lloyds Register Foundation; and Thomas Ashton Institute.

References

1. Abu-Jbara, A., Radev, D.: Coherent citation-based summarization of scientific papers. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 500–509. Association for Computational Linguistics (2011)
2. Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., Ha, V., et al.: Construction of the literature graph in semantic scholar. arXiv preprint arXiv:1805.02262 (2018)
3. Beltagy, I., Cohan, A., Lo, K.: Scibert: Pretrained contextualized embeddings for scientific text. arXiv preprint arXiv:1903.10676 (2019)
4. 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
5. Cohan, A., Goharian, N.: Scientific article summarization using citation-context and article's discourse structure. arXiv preprint arXiv:1704.06619 (2017)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
8. Jaidka, K., Chandrasekaran, M.K., Jain, D., Kan, M.Y.: The CL-SciSumm Shared Task 2017: Results and Key Insights. In: BIRNDL@ SIGIR (2). pp. 1–15 (2017)
9. Jaidka, K., Chandrasekaran, M.K., Jain, D., Kan, M.Y.: The CL-SciSumm Shared Task 2018: Results and Key Insights. In: BIRNDL@ SIGIR (2). pp. 1–15 (2018)
10. Jaidka, K., Chandrasekaran, M.K., Rustagi, S., Kan, M.Y.: Overview of the CL-SciSumm 2016 shared task. In: Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL). pp. 93–102 (2016)
11. Lebanoff, L., Song, K., Dernoncourt, F., Kim, D.S., Kim, S., Chang, W., Liu, F.: Scoring sentence singletons and pairs for abstractive summarization. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (2019)
12. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. R news **2**(3), 18–22 (2002)
13. Liu, Y.: Fine-tune BERT for Extractive Summarization. arXiv:1903.10318 [cs] (Mar 2019), http://arxiv.org/abs/1903.10318, arXiv: 1903.10318
14. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
15. Nomoto, T.: Resolving citation links with neural networks. Frontiers in Research Metrics and Analytics **3**, 31 (2018)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)

17. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). pp. 1532–1543 (2014)
18. Qazvinian, V., Radev, D.R.: Scientific paper summarization using citation summary networks. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. pp. 689–696. Association for Computational Linguistics (2008)
19. Radev, D.R., Muthukrishnan, P., Qazvinian, V., Abu-Jbara, A.: The acl anthology network corpus. Language Resources and Evaluation **47**(4), 919–944 (2013)
20. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
21. Soler, A.G., Apidianaki, M., Allauzen, A.: Word usage similarity estimation with sentence representations and automatic substitutes. arXiv preprint arXiv:1905.08377 (2019)
22. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic classification of citation function. In: Proceedings of the 2006 conference on empirical methods in natural language processing. pp. 103–110. Association for Computational Linguistics (2006)
23. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 4144–4150 (2017). https://doi.org/10.24963/ijcai.2017/579, https://doi.org/10.24963/ijcai.2017/579
24. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
25. Zerva, C., Batista-Navarro, R., Day, P., Ananiadou, S.: Using uncertainty to link and rank evidence from biomedical literature for model curation. Bioinformatics **33**(23), 3784–3792 (2017)
26. Zhu, J., Tian, Z., Kübler, S.: Um-iu@ ling at semeval-2019 task 6: Identifying offensive tweets using bert and svms. arXiv preprint arXiv:1904.03450 (2019)