Scientometrics (2020) 125:3109–3137
https://doi.org/10.1007/s11192-020-03455-z

Cited text span identification for scientific summarisation using pre-trained encoders

Chrysoula Zerva[1] · Minh-Quoc Nghiem[1] · Nhung T. H. Nguyen[1] · Sophia Ananiadou[1,2]

Received: 30 September 2019 / Published online: 7 May 2020

Abstract

We present our approach for the identification of cited text spans in scientific literature, using pre-trained encoders (BERT) in combination with different neural networks. We further experiment to assess the impact of using these cited text spans as input in BERT-based extractive summarisation methods. Inspired and motivated by the CL-SciSumm shared tasks, we explore different methods to adapt pre-trained models which are tuned for generic domain to scientific literature. For the identification of cited text spans, we assess the impact of different configurations in terms of learning from augmented data and using different features and network architectures (BERT, XLNET, CNN, and BiMPM) for training. We show that identifying and fine-tuning the language models on unlabelled or augmented domain specific data can improve the performance of cited text span identification models. For the scientific summarisation we implement an extractive summarisation model adapted from BERT. With respect to the input sentences taken from the cited paper, we explore two different scenarios: (1) consider all the sentences (full-text) of the referenced article as input and (2) consider only the text spans that have been identified to be cited by other publications. We observe that in certain experiments, by using only the cited text-spans we can achieve better performance, while minimising the input size needed.

Keywords Cited text span identification · Citation analysis · Scientific summarisation · Neural networks · BERT · Fine-tuning

✉ Chrysoula Zerva
chrysoula.zerva@manchester.ac.uk

Minh-Quoc Nghiem
minh-quoc.nghiem@manchester.ac.uk

Nhung T. H. Nguyen
nhung.nguyen@manchester.ac.uk

Sophia Ananiadou
sophia.ananiadou@manchester.ac.uk

[1] National Centre for Text Mining, School of Computer Science, University of Manchester, Manchester, UK

[2] Alan Turing Institute, London, UK

Introduction

Bibliometrics, citation analysis and scientific summaries are means that allow researchers to navigate scientific literature and process information faster and more efficiently. The exponential increase in scientific publications across fields renders the automation and improvement of such methods even more pertinent.

Within that realm, citation analysis approaches are being supported by machine learning algorithms to identify relations and impact of authors and publications (Ding et al. 2016). The identified citations can be interpreted as links to other publications, and are used to calculate a range of impact metrics between publications (Bornmann and Daniel 2009; Hutchins et al. 2016; Fister et al. 2016; Chang et al. 2019), which do not however capture the content of a citation. Improvements in text mining and natural language processing allow not only to identify citations in text but also to analyse the citation context, i.e., the text span that accompanies and explains a reference, describing the reasons for citing it. By analysing the citation context we can identify the motive for the citation (Teufel et al. 2006) as well as indirectly acquire information on the content of the referenced paper. Hence, assuming we have several citations to a specific publication, we could infer its main arguments, knowledge contributions or disputed claims.

However, the information we learn about a referenced paper is subjective and limited by the intentions, opinion and potential bias of the citing author(s). Instead, locating cited text spans in a referenced paper itself, i.e., finding the exact sentence(s) that are described by the citations, can provide us with more accurate information about the referenced knowledge. Being able to identify citing-cited text spans could aid in identifying the relation not only between publications, but also between specific arguments and pieces of information, thus allowing us to monitor the evolution of knowledge in a field. Identification of cited text spans also allows to better assess the impact, highlights and weaknesses of a referenced publication and could be used to improve citation-based metrics, as well as information retrieval and literature navigation tools (Hassan et al. 2018).

In this work we focus on the cited text span identification, treating it as a sentence pair classification task. We build on our approach presented in CL-SciSumm 2019 (Chandrasekaran et al. 2019; Zerva et al. 2019) examining in more detail the suitability and limitations of BERT (Devlin et al. 2019) for this task, compared to other architectures (including XLNET Yang et al. 2019; CNN Kim 2014; BiMPM Wang et al. 2017). The manually annotated dataset for this task is rather small, which constitutes a significant limitation in terms of training deep neural network models, especially since they were pre-trained on the generic domain. We thus explore the potential of fine-tuning the models using augmented or unlabelled data and compare the performance of domain-tuned to generic BERT language models. We evaluate the trained models using F-score as our primary criterion. To get more insights on the behaviour of each model we expand the evaluation using the mean reciprocal rank (MRR) (Craswell 2009) and an 'relaxed' F-score approach that allows us to consider the efficiency of each model in locating the wider area of text that a citation refers to Nomoto (2018).

In addition, we explore the potential value of cited text span identification for another important NLP task, namely, scientific summarisation. It has been claimed that since citations reveal the most important and mention-worthy information of a reference paper, their combination would capture all the paper's main points and contributions (Qazvinian and Radev 2008). This assumption motivated citation-based extraction of scientific summaries (Cohan and Goharian 2017; Abu-Jbara and Radev 2011). However, as discussed above, citing sentences might be more biased or noisy, compared to the corresponding cited text spans extracted from a reference paper itself (Jha et al. 2017). Hence, it has been proposed that cited text spans of the reference article could provide less biased information to support the scientific summarisation task (Cohan and Goharian 2017).

We intend to explore this scenario and compare the efficiency of using cited text spans for scientific summarisation, compared to using the full-text of the paper. To that purpose, we use an extractive summarisation approach, employing an adaptation of BERT which accounts for multi-sentential input (Liu and Lapata 2019). We examine two training configurations for the summarisation methods: (1) using the full paper sentences as input and (2) using only the combination of abstract and cited text spans. We evaluate the model using ROUGE-1, ROUGE-2, and ROUGE-L scores. The results show that in many configurations, performance is comparable between two scenarios on ROUGE-1 and ROUGE-L scores but using cited text spans consistently yields better ROUGE-2 scores.

Related work

CL-SciSumm shared tasks

We provide here the definitions of terms used throughout the article, related to the citation analysis (based on the CL-SciSumm tasks Chandrasekaran et al. 2019):

- *Full-text* The main body of a scientific publication, including the abstract but potentially excluding sections related to funding, acknowledgements, etc.
- *Citing paper (CP)* A full-text paper containing one or multiple citations to a reference paper
- *Reference paper (RP)* A full-text paper that is being cited by one or multiple citing papers.
- *Citance/citing sentence* A sentence that contains a reference to a specific paper.
- *Cited text span* The exact text span (sentence or word sequence) to which a citance refers.
- *Reference/cited sentence* A sentence in the RP which belongs to a cited text span.

The CL-SciSumm Shared Tasks (Jaidka et al. 2016, 2017, 2018; Chandrasekaran et al. 2019) are centred around supporting and promoting the identification of cited text spans and the subsequent use of those text spans for the generation of scientific summaries. The tasks build on the pilot TAC 2014 BioMedSumm task,[1] which was the first one to provide annotated resources with citing and cited sentences to support biomedical article summarisation. They propose a set of sub-tasks addressing the different steps that could lead to a more efficient scientific summarisation system, informed by cited text spans.

Since 2016 the challenges in CL-SciSumm (Chandrasekaran et al. 2019) are 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.

[1] TAC- Text Analysis Conference: https://tac.nist.gov/2014/BiomedSumm/.

- *Task 1A* For each citance, identify the spans of text (cited text spans) in the RP that most accurately reflect the citance.
- *Task 1B* 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.
- *Task 2* Generate a structured summary (of up to 250 words) of the RP.

Cited text span identification

In order to identify cited text spans, most work focuses around modelling the relation/relevance between a citing and a candidate cited sentence. Hence, several approaches for cited text span identification aim to model this relation by calculating textual similarity functions between citing and candidate cited sentences (Mihalcea et al. 2006), as a measure of relevance between the two. Early systems submitted to CL-SciSumm proposed similarity scores and features based on TF–IDF, latent semantic analysis (LSA) (Yeh et al. 2017; Cao et al. 2016; Nomoto 2016; Prasad 2017), or informativeness measures such as point-wise mutual information (PMI) (Yeh et al. 2017; Jha et al. 2017) and Jaccard similarity (Prasad 2017). Other systems used features based on the *n*-gram or sentence graph overlap (Aggarwal and Sharma 2016; Klampfl et al. 2016) in order to represent the similarity between the evaluated sentence pairs.

Other noteworthy approaches, focused on more specific modelling of the relation between cited and citing sentences, such as the adaptation of the Word Movers Distance (WMD) in combination with Latent Dirichlet Allocation (LDA) to infer the relevance between two text spans (Li et al. 2018). The generated features were then used to train supervised machine learning classifiers, including linear regression, tripartite neural networks, random forest (RF), and variations of support vector machine (SVM) classifiers. SVM and RF classifiers seemed to be better suited for the task, yielding high performance in the yearly challenges up to 2018 (Jaidka et al. 2018; Cao et al. 2016; Baruah and Kolla 2018; Li et al. 2017). Moreover, ensemble learning approaches and voting mechanisms applied on top of separately trained supervised classifiers, also seemed efficient on further improving the performance (Wang et al. 2018; Ma et al. 2018).

Nomoto (2016) advocated the use of neural networks and embeddings in order to model cited-citing sentence pairs. They initially proposed a combination of a TF–IDF model with a single layer neural network, which however did not seem to reach the performance of other supervised approaches. They then proposed the use of a triplet loss function (Wang et al. 2014) to train a neural network for the task and analysed the performance over different input embeddings. They also proposed the complimentary evaluation based on approximately correct targets (ACTs) where the goal is to find a region that likely contains a true target rather than its exact location (Nomoto 2018). Their approach on evaluation inspired the MRR and approximate PRF analysis presented in the results ("Results and discussion" section).

Apart from the aforementioned approach proposed by Nomoto, the continuous improvements in the field of deep learning and neural networks inspired more applications of such methods on this task. In CL-SciSumm 2018, there were submissions using embedding-based similarity measures (Baruah and Kolla 2018) and neural network architectures (CNN and LSTM) (Li et al. 2018; Abura'ed et al. 2018; Agrawal and Mittal 2018; De Moraes et al. 2018) to approach the task. However, due to the small dataset size, the best performing models were still based on RF and BM25 classifiers (Jaidka et al. 2018).

The 2019 shared task facilitated deep learning approaches by providing additional weakly supervised data for training (Chandrasekaran et al. 2019; Nomoto 2018). Indeed, while some approaches are still heavily based on hand-crafted features and similarity functions (La Quatra et al. 2019), there has been an increase both in the number and in the performance scores obtained by deep learning methods. Beijing university (Li et al. 2019) proposed the use of CNN combined with tailored Word2Vec feature maps to capture the weights for each sentence pairing, which demonstrated high F-score performance. Other approaches using CNN to calculate the validity of citing-cited sentence pairs have also been used Jha et al. (2017) and Ma et al. (2019). Siamese networks have also been employed for this task both as a standalone approach using a fully connected regression layer at the output (Fergadis et al. 2019), and in combination with other positional similarity features (Karimi et al. 2017). Overall, with the availability of augmented training data and the evolution of deep learning approaches and robust pre-trained embeddings, the application of deep learning techniques for efficient cited text span identification seems promising (Chandrasekaran et al. 2019).

In addressing this task, we noticed that it bears resemblance to a range of other NLP tasks, which assess the relation between spans of text. For example, sentence similarity, paraphrase extraction, question answering and inference identification are tasks that are often approached through identifying the relation between two sentences or passages. Bidirectional deep learning approaches that generate embeddings from sentence pairs to model relations between them, seem to be an efficient approach for such tasks (Devlin et al. 2019; Yang et al. 2019). Models such as BERT (Devlin et al. 2019), XLNET (Yang et al. 2019) and CTRL (Keskar et al. 2019) have been shown to produce good results for a wide range of sentence-pair tasks including question answering, machine translation and paraphrase. Matching network approaches such as Siamese networks (Nicosia and Moschitti 2017; Neculoiu et al. 2016), Bilateral Multi-Perspective Matching (BiMPM) (Wang et al. 2017) and other matching network proposals (Duan et al. 2018; Nie and Bansal 2017), also seem promising for tasks related to sentence pairing. However, with the exception of Siamese networks such models have not been used for cited text span identification. We hence decided to experiment with methods based on BERT encoders (Devlin et al. 2019) and also compare with aforementioned architectures to evaluate their suitability and adaptability to the task.

Scientific summarisation

Scientific document summarisation is a well researched field. The study conducted by Luhn (1958) is the very first work on technical paper summarisation using a statistical-based approach. His approach firstly detects descriptive words based on the frequency of occurrence in a document. The summary is then created by selecting sentences with high density of descriptive words. Later work based on this idea proposed multiple features to indicate sentence importance (Edmundson 1969), weights for words (Conroy et al. 2006), or statistical tests on word distributions (Lin and Hovy 2000).

Several studies have applied different machine learning techniques for summarisation. Typically, machine learning approaches treat summarisation as a classification task in which a sentence is classified into two classes: included or not included in the summary. The work by Kupiec et al. (1999) was among the first studies that used machine learning techniques for summarisation. They trained a Naive Bayes classifier with several features including the location and the length of a sentence, fixed phrases, the position of paragraphs and word frequency features. Other work showed that machine learning methods that exploit the dependency between sentences (e.g., Hidden Markov model Conroy and O'leary 2001 and Conditional Random Fields Shen et al. 2007) often outperform other techniques.

Recently, deep learning-based approaches have been applied to the summarisation task and achieved remarkable results. SummaRuNNer (Nallapati et al. 2017) is a Recurrent Neural Network-based sequence model for extractive summarisation of documents. Zhou et al. (2018) proposed a novel end-to-end neural network framework for summarisation by jointly learning to score and select sentences. The work of Rush et al. (2015) was among the first studies that have attempted to use sequence-to-sequence models for abstractive summarisation. Up to now, several studies have enhanced the sequence-to-sequence model with copy mechanism, coverage model (See et al. 2017), and reinforcement learning (Paulus et al. 2017).

Inspired by recent work on pre-training of deep bidirectional transformers for language understanding (Devlin et al. 2019), some authors have considered the use of pre-trained encoder for document encoding in summarisation. Liu et al. (2019) proposed a model that induces a multi-root dependency tree while predicting the output summary. Liu and Lapata (2019) introduced a novel document-level encoder based on BERT and achieved state-of-the-art results on three datasets. Zhang et al. (2019) proposed the hierarchical BERT to pre-train document level encoders on unlabelled data. Miller (2019) has utilised BERT for extractive text summarisation on a python-based RESTful service for lecture summarisation.

The aforementioned approaches have shown significant improvement in terms of performance in automated summarisation, but they are implemented and optimised mainly for application on newswire datasets (CNN, DailyMail, NYT etc.). While the intention is similar the document characteristics differ between the newswire and scientific domain. Apart from the domain specific language and document structure, there is an important difference in terms of the length of documents and produced summaries between the two fields. Typically the expected summary size in the newswire domain is approximately 50 words long, while the corresponding scientific summaries range between 150 and 250 words. At the same time, the length of documents to be summarised also differs: The average document length in newswire corpora is approximately 700 words while for full-text scientific publications it is approximately 4000 words (specifically the documents in CL-SciSumm corpus are approximately 6.3 times longer than the ones in CNN/DailyMail corpora).

In extractive summarisation models, the length of the produced summary can be adapted by controlling the stopping criteria in algorithms such as Maximal Marginal Relevance (MMR) (Carbonell and Goldstein 1998) which are used to select the sentence to be added in the final summary. Yet, this does not alleviate the issue arising from the differences in the document length and structure. To deal with the length of scientific summaries most approaches focus on identifying the characteristics of useful sentences. Related work has focused on argumentative zoning identification and scientific discourse (Contractor et al. 2012; Cohan and Goharian 2017), identification of citing sentences and their function (Saggion et al. 2016; Mohammad et al. 2009) as well as identification of cited text spans (Galgani et al. 2015) have all been proposed as methods to limit the scope of input for the models in an informative way. Additionally, abstracts in scientific papers, typically contain important information that presents the main contributions of a paper. Xu et al. (2015) proposed a statistical framework to extract a scientific summary from a collection of citations to a paper, which they referred to as "citation summary". Other researchers also utilised citation summaries to that end (Mollá et al. 2014; Galgani et al. 2012). Recently,

Table 1 Dataset statistics for cited text span identification

	2019	2019-AUG
Positive instances	1154	30,407
Negative instances	3058	66,089
Avg words per CP text	46	35
Avg words per RP text	29	28
Avg ref sent per citation	1.72	1.00

cited text spans are also used for the same purpose. Galgani et al. (2015) combined citing and cited text spans in a form of a citation network to extract summaries in the legal domain and showed that their method outperformed competitive baselines. Several participating systems in the CL-SciSumm shared-tasks have explored various methods for creating summaries based on cited text spans, such as word embeddings, sentence clustering, CNN-based learning, and sequence-to-sequence generating (Jaidka et al. 2016, 2017, 2018; Chandrasekaran et al. 2019). Together these studies provide important insights into the scientific summarisation task and motivate our approach. We choose to focus on examining the suitability of the abstract and cited text spans when used instead of the full-text as input to BERT based architectures that have proved efficient in newswire summarisation. We note that if we only take into account the abstract (approximately 150 words) and the cited sentences (approximately 500 words) of a scientific paper, we will obtain a relatively similar length to the newswire corpora (approximately 700 words), so we will not deviate significantly in terms of input size.

Data resources and pre-processing

Cited text span identification

CL-SciSumm datasets

For all experiments presented in this work we used the data provided for the CL-SciSumm 2019 Shared task (Chandrasekaran et al. 2019). For cited text span identification, the organisers provided two different datasets for training: (1) a manually annotated dataset comprising 40 articles and their respective citing papers, which were also used in the 2018 CL-SciSumm challenge, and (2) an augmented dataset of 1000 articles and their respective citing papers, which were automatically annotated with a neural network approach as described in Nomoto (2018). Henceforth, we will refer to the first dataset as the *2019 CL-SciSumm dataset*, the second one as the *2019-AUG dataset*.

Note that the 2019 dataset may contain several consecutive sentences for each annotated citation in the CP and may correspond to multiple, not necessarily consecutive, cited text spans in the RP. The 2019-AUG dataset contains strictly one citation sentence and one cited sentence for each instance. The statistics for the two datasets are presented in Table 1.

All CL-SciSumm data is organised as one text file per annotated reference paper (RP) and the information about the citing - cited text spans is provided using the following format:[2]

> Citance Number: 15 | Reference Article: *C02-1025.xml* | Citing Article: *W03-0423.txt* | Citation Marker Offset: ['12'] | Citation Marker: *Chieu and Ng, 2002b* | Citation Offset: ['12'] | Citation Text: <S sid = "12" ssid = "12"> *Such global features enhance the performance of NER (Chieu and Ng, 2002b).*</S> | Reference Offset: ['4'] | Reference Text: <S sid ="4" ssid = "4">*In this paper, we show that the maximum entropy framework is able to make use of global information directly, and achieves performance that is comparable to the best previous machine learning-based NERs on MUC6 and MUC7 test data.*</S> | Discourse Facet: *Results_Citation* | Annotator: *Aakansha Gehlot* |

We processed this information to generate positive training and testing instances (sentence pairs between citing sentences and text spans from the RP). The CL-SciSumm data is generated using a subset of the ACL anthology reference corpus (Radev et al. 2013), which in turn was generated by OCR. As a result, there are erroneous words and erroneously segmented sentences in the annotations. We applied a set of sentence reconstruction rules to the RP and CP sentences to correct segmentation errors such as erroneous sentence breaks after parentheses, enumeration or abbreviations. The coprus pre-processing script is made available as an ipython notebook[3] and the details and statistics of the OCR errors addressed are described in the "Appendix 1".

For the generation of negative instances, 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 to filter noisy sentences. The filtering methods are described in detail in the "Appendix 1". In order to keep a balance between adequate training data and labels, we chose a proportion of 1:4 positive to negative pairs per citance. The same processing is applied on both the 2019 and 2019-AUG dataset.

We evaluate our methods using tenfold cross validation on the 2019 dataset, since it is the only dataset that was manually annotated to a gold standard.

ACL anthology reference corpus (ACL-ARC)

The ACL Anthology reference corpus (Bird et al. 2008) was used in order to experiment with fine-tuning the BERT language model on a domain specific dataset, prior to training on the CL-SciSumm data.

We used the v2.0 version (March, 2016) and the files processed and formatted by the ParsCit software (Councill et al. 2008). We further processed each document to retain only passages marked as < *bodytext* >. These text passages were filtered to remove noisy ones, as described in the "Appendix 1". The resulting corpus which was used for fine-tuning to the target domain amounted to 7,205,084 sentences.

[2] Text in italics corresponds to variables—annotated information.

[3] https://github.com/chryssa-zrv/cited-text-span-id/.

Scientific summarisation

For training and evaluating our models we used two benchmark datasets, namely the CL-Scisumm 2019 dataset and the ScisummNet dataset (Yasunaga et al. 2019). The CL-SciSumm 2019 dataset has 40 research papers randomly sampled from the ACL Anthology reference corpus. Note that the CL-SciSumm 2019 dataset for scientific summarisation contains the same 40 papers that are manually annotated in the 2019 CL-SciSumm dataset used for the cited text span identification and described in the previous section.

The ScisummNet dataset contains the 1000 most cited papers from the same ACL Anthology reference corpus and the respective human-written summaries. The 1000 papers have 21–928 citations in the anthology. They also fully overlap with the 1000 papers used as the augmented 2019-AUG dataset. For the generation of the human-written summaries, the 1000 papers were treated as RP. From their respective CP, 20 citances for each paper were selected and provided to the annotators along with the abstracts. Thus the annotators produced the summaries without reading the full-text.

For the experiments presented below we removed overlapping papers with the CL-Scisumm 2019 test dataset, resulting in a total of 978 papers. In all summarisation experiments, we use the ScisummNet dataset for training and report the testing result on the CL-Scisumm 2019 dataset.

In order to prepare the data for this task, we firstly filter out too long (more than 45 tokens) or too short (less than 5 tokens) sentences. Any unrelated sentences, i.e., sentences that belong to “Acknowledgment” or “References” sections, are also removed. We then tokenise the text using the stanford-corenlp toolkit.[4]

As explained before, the training data was created using abstractive summarisation methods, i.e., the annotators produced their own sentences rather than copying sentences from original papers. These summaries, however, cannot be directly used to train extractive summarisation models. Hence we pre-process the summaries to create an extractive summary version of the originally provided data. To identify which sentences should be put into an extractive summary, we greedily selected sentences that maximise the ROUGE scores. To generate training data for the classifier (described in “Methods” section), we assigned label 1 to sentences selected in the extractive summary version and 0 otherwise, thus obtaining positive and negative instances.

Overview of approach

In the following sections we describe our methods and results for the cited text span identification task (“Cited text span identification approach” section) and the scientific summarisation task (“Scientific summarisation” section).

Regarding the cited text span identification task, we fine-tuned the BERT model (next sentence prediction classifier) for the CL-SciSumm 2019 task. We compared it against BERT models fine-tuned on domain specific data first, as well as other architectures (XLNET, Convolutional Neural Network (CNN), and Bilateral Multi-Perspective Matching (BiMPM)).

Meanwhile, for the scientific summarisation task, we also fine-tuned the BERT model and experimented with three different types of output layers: linear, Bidirectional Long Short-Term Memory (BiLSTM), and bidirectional transformer.

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

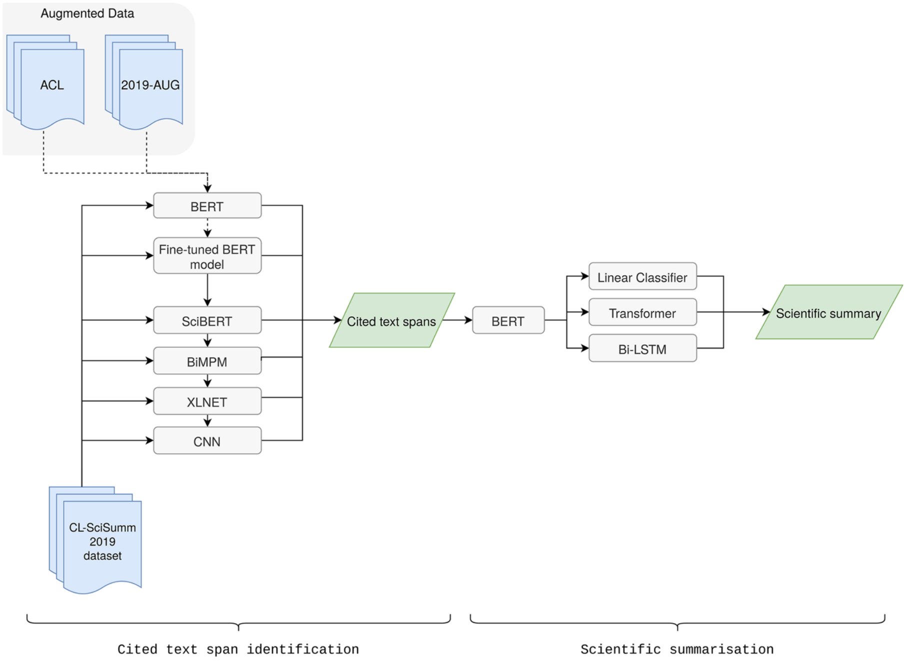


Fig. 1 An overview of the methods and DNN models used and compared against each other (grey rounded components) for cited text span identification and scientific summarisation

Table 2 Example of valid and invalid citation pairs, extracted from document C04-1089 of the 2019 CL-SciSumm corpus

	Citing sentence	Reference sentence
VALID	Shao and Ng (2004) presented a method to mine new translations from Chinese and English news documents of the same period from different news agencies, combining both transliteration and context information	In this paper, we propose a new approach for the task of mining new word translations from comparable corpora, by combining both context and transliteration information
INVALID	Shao and Ng (2004) presented a method to mine new translations from Chinese and English news documents of the same period from different news agencies, combining both transliteration and context information	While we have only tested our method on Chinese–English comparable corpora, our method is general and applicable to other language pairs

An overview of the methods we compare and the datasets used is depicted in Fig. 1.

Cited text span identification approach

Methods

We treat the task as a sentence pair classification task, where we have to classify a citing sentence and a candidate reference sentence as a valid or invalid pair. Within the context of this task, a valid pair corresponds to a citing sentence that refers to the idea mentioned in the reference sentence of the pair. Hence all our models are trained to output probabilities for each label "INVALID" or "VALID", which are encoded as 0 and 1, respectively. An example is presented in Table 2.

Inspired by performance on other text matching tasks we experiment with BERT (Devlin et al. 2019) as our primary model. Specifically, since BERT is pre-trained on a *language modelling* (LM) and a *next sentence prediction task*, its architecture and learned embeddings can readily account for textual sequence pairs and be adapted to citing-cited sentence pair identification.

Our main approach uses the *bert-base-uncased model*, with the following setup: 12 layers, hidden vectors of size 768 and 12 self-attention heads.[5] We specifically adapted the "NextSentencePrediction" pre-trained model and fine-tuned on the CL-SciSumm 2019 dataset. We compare that to the use of other network architectures as well as the use of augmented data. We refer to this as the *BERT-base model* henceforth.

Fine-tuning on augmented data

The fact that BERT was pre-trained on data from the general domain, could render it sub-optimal for application on tasks pertaining to scientific text. Hence, we investigate the potential of using unlabelled or machine generated data to fine-tune the pre-trained BERT model prior to training on the 2019 dataset. We explore three different approaches using augmented data to improve performance:

1. *Using a BERT model pre-trained on domain-specific data*: We use the SciBERT model (Beltagy et al. 2019), which is pre-trained on a collection of 1.14M documents from Semantic Scholar (Ammar et al. 2018). Specifically the collection consists of 18% papers from the computer science domain and 82% from the broad biomedical domain. Assuming that the vocabulary of the SciBERT model (*scibert-scivocab-uncased*) is closer to the task, we compared its performance to BERT (without further fine-tuning).
2. *Using large, unlabelled, domain-specific data*: Fine-tune the weights of the pre-trained BERT model on the ACL-ARC (Radev et al. 2013) and then train on the CL-SciSumm data as above. We henceforth refer to the resulting model as the *ACL model*.

   For fine-tuning on the ACL-ARC corpus, we use the next sentence prediction configuration. In each epoch we choose whether to sample a pair of consecutive or random sentences with 0.5 probability respectively. We employ the fine-tuning approach described in (Howard and Ruder 2018) and fine-tune for 3 epochs and a batch size of 16, with an initial learning rate of $LR = 3e - 5$.
3. *Using augmented, task-specific data*: as augmented data, we refer to a dataset that was automatically generated either with distant supervision or after training a model on a specific task. In this case, we use the augmented data provided by the CL-SciSumm shared task 2019 (see also "Data resources and pre-processing" section). We compared the following approaches:

[5] For all experiments that use BERT models, the relevant code is implemented in Python, using the Pytorch library. All BERT pre-trained models are provided by https://github.com/huggingface/transformers, version 2.4.0, which has been verified to reproduce the outputs of the original TensorFlow implementation.

(a) Fine-tune the weights of pre-trained BERT as described for (2), but instead of consecutive and random sentences from unlabelled data we use citing-cited pairs that have been extracted automatically versus random pairs (denoted as 19-AUG-FT in the results).

This way we intend to examine whether using the same fine tuning approach with a smaller but task specific corpus would provide us with a better model. While the 2019-AUG dataset is significantly smaller than the ACL-ARC corpus, the automatically generated sentence pairs are closer to the original task, and thus could compensate for the corpus size.

(b) Combine the augmented data with the manually labelled data and train BERT on the combined corpus, with the same configurations as for the BERT-base model (denoted as 19-AUG in the results).

(c) Use positive instances from the 2019-AUG dataset, train for 5 epochs and then revert to training on the manually annotated 2019 dataset with weight and learning rate decay (denoted as 19-AUG-P in the results).

Note that we apply 19-AUG (3b) and 19-AUG-P (3c) approaches on the BERT, SciBERT and ACL models as well.

Comparison with other architectures

We compared the BERT base model to the following architectures.

CNN with BERT features: In this case we experimented with the use of a CNN layer that takes the output of the BERT pre-trained model as input features. To produce a feature vector for each token, we initially concatenated the last four layers of the BERT model and used 2×2 MaxPooling (Zerva et al. 2019), which has been shown to achieve optimal performance according to Devlin et al. (2019). Here we opted for maintaining the 4×768 dimension instead of concatenation. We experimented both with the BERT and SciBert embeddings. The CNN architecture consists of three convolution layers, followed by a fully connected linear layer. We use 3×3 MaxPooling after each convolution layer and a dropout of 0.1 after the last convolution layer.

We initially experimented with the use of the full sentence embeddings against the use of the [SEP] and [CLS] embeddings only, which capture the sentence and sentence pairing information, respectively (see Fig. 2). As shown in the "Appendix 3" we found that using only the [SEP] + [CLS] embeddings results in better performance, while it consumes less memory and running time compared to using the full sentence embeddings. Hence, in Table 5 we compare the BERT base model with the CNN+BERT and CNN+SciBERT approaches, which both use the [SEP]+[CLS] embeddings as input.

We also experimented with the use of additional features, such as the position of a sentence in a document and the section of the publication (sid and ssid offsets). However, the experimental results showed that these features did not help the performance. We therefore present those results in the "Appendix 3".

XLNET: We compare the performance of BERT to the use of XLNET (Yang et al. 2019). XLNET has been shown to outperform BERT in several NLP tasks, addressing one of BERT's weaknesses in the language modelling approach. Instead of masking inputs, XLNET uses an auto-regressive method, and learns bidirectional contexts by maximising the expected likelihood over all permutations of the factorisation order. It has been shown to outperform BERT on various tasks, including several paraphrasing tasks of the GLUE

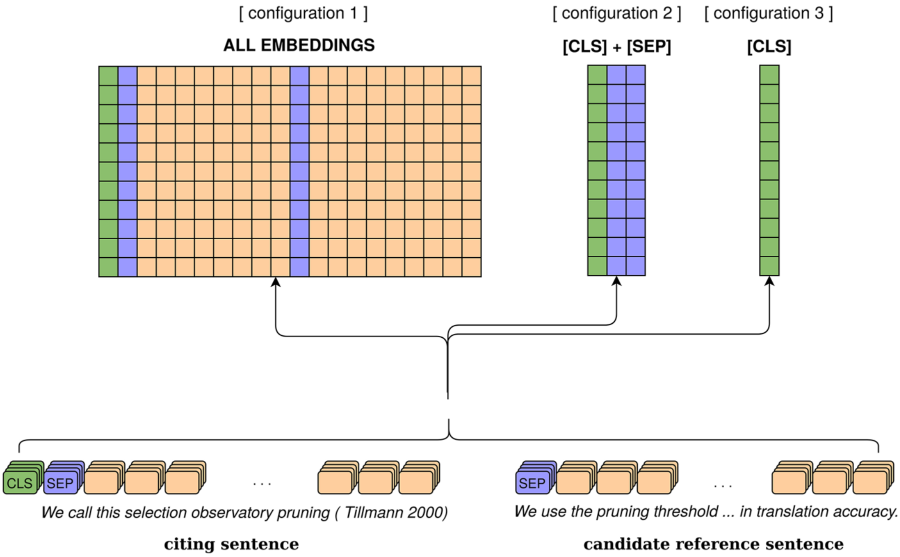

[configuration 1]
ALL EMBEDDINGS
[configuration 2]
[CLS] + [SEP]
[configuration 3]
[CLS]
CLS
SEP
...
We call this selection observatory pruning (Tillmann 2000)
citing sentence
SEP
...
We use the pruning threshold ... in translation accuracy.
candidate reference sentence

Fig. 2 Different sentence embedding configurations to be used as input features for the CNN

benchmarks as well as question answering tasks (SQUAD) (Rajpurkar et al. 2016), RACE (Lai et al. 2017). To repeat the experiments with XLNET, we used the *xlnet-base-cased* model provided on huggingface transformers (Wolf et al. 2019) for Pytorch. For the hyperparameter setup, we used the fine-tuning configurations provided for the SQUAD dataset.

Bilateral multi-perspective matching model: With the intuition that there is probably some correlation or shared information 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) (Wang et al. 2017) to identify cited text spans. 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-trained embeddings (Pennington et al. 2014) as input to BiMPM. We then conducted experiments with 100 epochs and a batch size of 6. Similar to the CNN approach, we also experimented with feature vectors extracted from BERT. Specifically, we replaced Glove embeddings by vectors resulting from 2 × 2 MaxPooling over the last four BERT layers. To address the issue of imbalanced training data, we set different weights for positive and negative pairs in the loss function, including 0.4 versus 0.6 and 0.3 versus 0.7. Even with re-weighting, the model could not learn to distinguish positive pairs; we therefore only report results on the GloVe embeddings.

Results and discussion

Evaluation setup

Following the evaluation metrics used in CL-SciSumm 2019 shared task, we use Precision, Recall and F-score as the primary performance metrics, to evaluate the predictions against

Table 3 Performance on the 2019 dataset for the BERT base model and k ranging from 1 to 5

k	Recall	Precision	F-score
1	**0.173**	0.113	0.137
2	0.162	0.192	0.176
3	0.149	0.259	**0.189**
4	0.133	0.299	0.184
5	0.121	**0.337**	0.178

Bold values correspond to the highest obtained performance for each metric

the gold label annotations. For these metrics, the evaluation is considered on a sentence level (we use the sentence ids indicated as correct to compare against the predictions).

Apart from those metrics we also want to get a better understanding of the distance between predictions and the gold labels (focusing on the case of false positives). Distance in this case can be estimated indirectly via:

1. *MRR* By estimating the mean reciprocal rank (MRR) for each classifier we can obtain a measure of where a correct (gold label) sentence falls based on the range of predicted scores. Thus we can discriminate between models that still attribute high probabilities for correct citing-cited pairs and those that have a behaviour closer to random.

   We compare the MRR performance for all systems discussed above. We note that a classifier might attribute the same score to several sentence pairs. Thus we consider two types of MRR scores, namely group rank (MRR-g) and no group rank (MRR-ng). Regarding MRR-g, we group pairs that are attributed same probabilities by the model into the same rank. For MRR-ng, we randomly permute the order of sentence pairs with same scores when evaluating the output. For example, if we have two sentence pairs sp_1 and sp_2 and the model attributes both of them with probability p_i, which is the highest probability score above all pairs, the pairs would be attributed a rank r_i as follows:

   - group-rank: $r_1 = r_2 = 1$;
   - no group-rank: $r_1 = 1, r_2 = 2$ if the random probability generation is $rpg > 0.5$; otherwise $r_1 = 2, r_2 = 1$.

2. *Relaxed PRF using ACT* In this case we consider predicted cited sentences correct if they fall within a varying window (window size n ranging from 1 to 5) around the gold annotated label and repeat the Precision, Recall and F-score (PRF) evaluation. This evaluation approach is inspired by Nomoto (2018) who argues that examining a wider cited area is important in getting a broader picture of how such models perform. They refer to this approach as *approximately correct targets (ACT).*

We use 10-fold cross validation on the 2019 dataset for all reported results below. We also present the performance on the 2019 test dataset for CL-SciSumm shared task in the "Appendix 2" as a point of reference.[6] For each fold, the results for Precision, Recall and F-score were generated by the following procedure. We firstly generate all possible citing-cited sentence pairs for each RP and then apply the trained models on each pair. We rank the output for each pair based on the score for the 'VALID' class, and choose

[6] The performance on 2019 CL-SciSumm test data was calculated by the CL-SciSumm 2019 organisers and we do not have access on the gold labels to further analyse those results.

Table 4 Performance on the 2019 dataset for the BERT base model and the fine-tuned approaches

Model	Precision	Recall	F-score	MRR-ng	MRR-g
BERT-base 2019	0.149	0.259	0.189	0.285	0.697
BERT-base 19-AUG	0.144	0.277	0.189	0.156	0.633
BERT-base 19-AUG-P	0.168	0.265	0.206	0.165	0.687
ACL	0.159	0.289	0.205	**0.293**	0.747
ACL 19-AUG	0.155	0.255	0.193	0.179	0.751
ACL 19-AUG-P	**0.172**	**0.277**	**0.213**	0.181	**0.752**
SciBERT 2019	0.118	0.220	0.154	0.252	0.695
SciBERT 19-AUG	0.085	0.101	0.092	0.125	0.789
SciBERT 19-AUG-P	0.103	0.156	0.135	0.185	0.658
19-AUG-FT	0.148	0.271	0.191	0.192	0.688

Bold values correspond to the highest obtained performance for each metric

the top k pairs to return as positive instances. We initially evaluate the BERT base model on 2019 dataset for $k = [1, 5]$ as shown in Table 3 and report results for $k = 3$ for the rest of the configurations, since we obtain the best F-score value for this k.[7]

Fine-tuning on augmented data

In Table 4 we present the performance for the BERT models, firstly using BERT-base on 2019 CL-SciSumm dataset and subsequently for the different fine-tuning combinations mentioned in "Fine-tuning on augmented data" section.

We can observe that fine-tuning the BERT-base model on the ACL-ARC corpus prior to training on the 2019 dataset yields the best performance and outperforms both BERT-base, SciBERT, and the 19-AUG-FT model. In comparison, the 19-AUG-FT approach does not perform as well as the ACL one. We identify two potential reasons for the difference in performance: (1) the limited size of the 2019-AUG augmented data compared to the ACL-ARC corpus and (2) the noise in the generation of the 2019-AUG data. The noisy data could explain the low performance when combining the 2019 dataset with the 2019-AUG one. Indeed, the automatically generated pairs for the 2019-AUG dataset, are limited to one reference sentence per citation. We have seen that the 2019 manually annotated dataset often has multiple reference sentences paired to each citation sentence (>20%), thus we can assume that the same would be true for the 2019-AUG data, hence this would account for a portion of false negatives. We tried to alleviate this by experimenting with the use of only the positive instances, but that approach also proved to be insufficient. With the 19-AUG-P approach however, i.e., when we used only the positive instances for the first 5 epochs, and then continued training only on the 2019 CL-SciSumm dataset with decaying learning rate, we can see that both the ACL and the BERT models get an improvement in performance. The combined ACL + 19-AUG model obtains better performance, indicating that the combination of the two types of domain specific datasets is beneficial. On the other hand, we noticed that in all those models the ability to discriminate between valid/invalid instances in terms of the predicted score, actually worsened (leading to high MRR-g scores and low MRR-ng scores). In fact, when applied on the SciBERT model, the classifier predicted very similar and low scores for all pairs, leading to reduced F-score performance.

[7] We noticed similar patterns for the rest of the algorithms tested.

Table 5 Performance on the 2019 dataset for the XLNET, CNN and BiMPM models

Model	Precision	Recall	F-score	MRR-ng	MRR-g
BERT-base	**0.149**	0.259	0.189	**0.285**	0.697
XLNET	0.104	0.188	0.134	0.225	**0.731**
CNN+BERT	0.145	0.263	0.187	0.260	0.685
CNN+SciBERT	0.148	**0.271**	**0.191**	0.265	0.694
BiMPM	0.035	0.070	0.047	0.096	0.315

Bold values correspond to the highest obtained performance for each metric

We intend to further experiment with augmenting data in future work, looking into distant learning and instance re-weighting approaches that could help us take better advantage of such domain-specific resources.

Comparison with other architectures

In Table 5 we present the comparison with other architectures for the 2019 dataset.

With the exception of CNN+SciBERT no other model outperforms the BERT-base approach. In terms of the XLNET architecture, this could be owed to the fact that XLNET did not use the next sentence prediction task during the training of the language model. While in the XLNET paper it is specified that using the next sentence prediction for pre-training did not add any significant performance boost, perhaps in this specific task the information captured by the BERT language model using the next sentence prediction task is beneficial. We note however, that XLNET obtains a high score for the MRR-g metric, indicating that part of its performance drop is due to the fact that many pairs obtain the same high-positive score (i.e., the model does not learn to discriminate between different pairs that well). In interpreting the results, it has to be noted that we experimented only with the hyper-parameters specified in the supplement of the XLNET paper and optimised only for the number of epochs. Thus, further experimentation could lead to improved performance.

The CNN model used with BERT embeddings as input features, obtained comparable performance, even surpassing BERT-base when using the SciBERT embeddings. This result opposes the performance of SciBERT presented in Table 4. Since in the initial architecture the [CLS] embedding is used for the final prediction, while in this CNN configuration we use [CLS]+[SEP] embeddings, we speculate that SciBERT captures more accurately the sentence information, hence the performance boost. We present a further analysis of the CNN configurations in the "Appendix 3" and we intend to further explore the potential of the CNN for this task in future work.

Relaxed PRF using ACT

As pointed out by Nomoto (2018), ACTs allow us to consider the efficiency of the models in a less strict citation resolution task. In Fig. 3 we can observe the F-score performance of a selection of the presented approaches on the 2019 dataset. We noticed that for most models performance increases at a comparable rate as we widen the window of valid ACTs. Interestingly, the increase for the 19-AUG-FT model (purple line, with triangle marks) is significantly greater and for $w \geq 2$ outperforms all other models. We attribute this to the

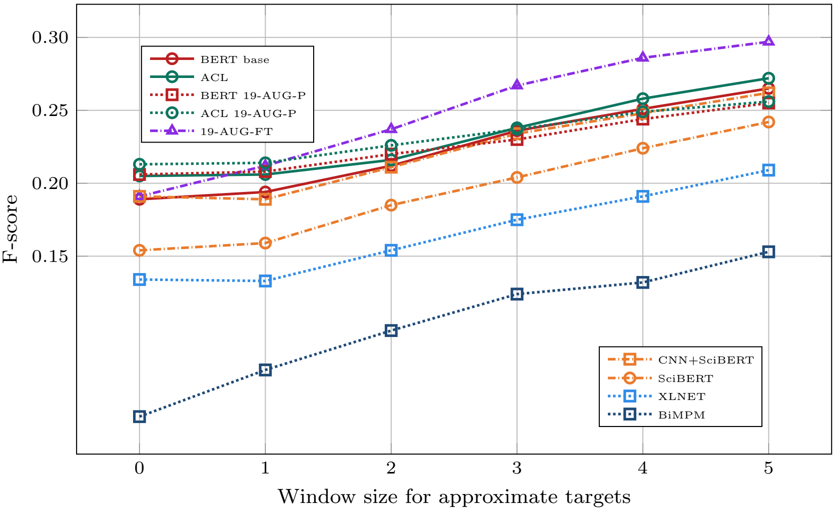


Fig. 3 Relaxed F-score performance for a window size up to 5 for on the 2019 CL-SciSumm dataset

Table 6 The number of RP sentences that are repeatedly cited

Dataset	2 times	>2 times
2019	162	170
2019-AUG	2996	2001

fact that the 2019-AUG data used for fine-tuning that model was generated with a method that aims to optimise performance in the "approximate" targets rather than the exact annotated data. This observation could help alleviate the impact of noise from the augmented 2019-AUG data, discussed in the previous sections, and incorporating ACT's in the training process could aid to achieve better performance.

Discussion on multi-cited text spans

Our observations on the training set show that RP sentences are repeatedly cited from different CP citing sentences. Table 6 shows that half of the RP sentences are cited twice while the others are cited from 3 to 17 times in the 2019 dataset. We observed that this fact might bias our models in favouring specific sentences, but it also significantly affects the calculated performance in the case of missing sentences of the RP that are frequently cited.

As shown in Table 7 all models are prone to predicting the same RP multiple times. However, there appears to be a direct relation between the performance of the model and the percentage of repeated predictions. Specifically, it seems to be one of the main obstacles in the performance of the BiMPM model, since it consistently favoured the same RP sentences. Also, comparing the BERT-base model to the fine-tuned versions (ACL, SciBERT and 19-AUG-FT), we can see that fine-tuning helps to avoid the repetitive RP predictions. Even in the case of the 19-AUG-FT model, which has lower F-score performance than BERT-base, the proportion of repeated RP sentences in the predictions is significantly lower. The same pattern, although less pronounced, we can observe between the CNN+BERT and CNN+SciBERT models.

Table 7 Proportion of repeated RP predictions in each model configuration for the 2019 CL-SciSumm dataset

	1	2	3–5	6–10	11–20	>20
BERT-base	0.59	0.18	0.16	0.06	0.02	0.00
ACL	0.60	0.20	0.15	0.04	0.01	0.00
19-AUG-FT	0.57	0.20	0.17	0.05	0.01	0.00
BERT 19-AUG-P	0.64	0.17	0.14	0.03	0.01	0.00
ACL 19-AUG-P	0.64	0.18	0.13	0.03	0.01	0.00
SciBERT	0.63	0.19	0.14	0.03	0.01	0.00
XLNET	0.55	0.18	0.19	0.06	0.01	0.00
BiMPM	0.40	0.20	0.18	0.13	0.06	0.03
CNN+BERT	0.57	0.14	0.18	0.08	0.03	0.01
CNN+SciBERT	0.58	0.16	0.13	0.09	0.03	0.01

Scientific summarisation

Methods

We formulate the summarisation task as a binary classification problem as well. The classifier needs to classify sentences provided as input into two classes: *included* or *not included* in the summary. The trained model outputs a probability for each class and we can then rank sentences based on how likely they are to be *included* in the final summary. From the ranked list, we add sentences into the final summary one by one, using Maximum Marginal Relevance (MMR) and ensuring that there is no tri-gram overlap between the current summary and the sentence to be added. In this way, we avoid adding redundant sentences with very similar content, which would not add new information to the summary. The process stops when the summary reaches the predefined, maximum length (i.e., 250 words in this task).

Following the previous tasks, we also employ BERT for scientific summarisation. Specifically, the BERT-based classifier that we use is similar to the one by Liu (2019). We maintain sentence vector encoding of BERT by using the `[SEP]` embeddings to capture features for each sentence. However the `[CLS]` symbols are re-purposed and used to signify the beginning of each sentence. Hence, in order to model multiple sentences, we capture features for all sentences ascending each `[CLS]` symbol. An odd sentence is assigned a segment embedding E_A while an even sentence is assigned a segment embedding E_B. Finally, a linear layer is added to the BERT output to predict a score for each sentence (1 is *included,* 0 is *not included*). Besides the linear layer, we also experiment with a bidirectional transformer layer and a Bidirectional Long Short-Term Memory (Bi-LSTM) layer.

The small size of the CL-SciSumm dataset rendered it harder to train a deep neural model. To address this issue, we train all of our models using the data from SciSummNet (Yasunaga et al. 2019). 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 papers, which makes them all subjective sentences. We therefore apply simple heuristics (for example, change “our” to “their”) to convert these subjective sentences to objective ones after generating a summary.

Table 8 ROUGE F1 results on CL-SciSumm 2019 data

Setup	Method	ROUGE-1	ROUGE-2	ROUGE-L
Full text	Linear	**47.85**	22.50	**45.32**
	Linear*	44.12	19.65	41.75
	Transformer	47.49	21.58	44.86
	Transformer*	43.90	19.39	41.62
	Bi-LSTM	47.64	21.38	44.88
	Bi-LSTM*	43.64	19.18	41.33
Abstract + Citances	Linear	47.19	**24.78**	44.47
	Linear*	45.57	24.46	43.06
	Transformer	46.16	23.54	43.36
	Transformer*	44.51	23.15	41.95
	Bi-LSTM	46.84	24.18	43.96
	Bi-LSTM*	45.03	23.81	42.47
Other systems	BertSumExt	41.66	22.34	38.93
	BertSumExtAbs	41.33	20.80	38.48

Bold values correspond to the highest obtained performance for each metric

*indicates systems with augmented abstract

Results and discussion

Evaluation setup

Regarding the scientific summarisation task, we evaluated our systems by calculating ROUGE-2 score (Lin 2004) when matching the generated summaries against the provided summaries by the CL-SciSumm 2019 shared task. We use the ROUGE-2 score as the main determinant for performance since it is the most commonly used one and it has been claimed to have better accuracy for summarisation tasks (Lin 2004). We complement the evaluation using ROUGE-1 and ROUGE-L scores, in order to get a comprehensive overview of the performance for each model.

All models use *bert-base-uncased* model with 50,000 training steps on a single GTX 1080Ti GPU. To prevent over-fitting, we set the dropout to 0.1. Learning rate is 0.002 with warming-up on first 10,000 steps to reduce the primacy effect of the early training samples. The models accumulate the gradients every two steps.

Results

Table 8 shows the results of our models on different settings. The first and second blocks of the table show the results where the models select sentences from full papers and from a combination of abstracts and citances, respectively. Following Yasunaga et al. (2019), we also present the results with augmented abstract setting where the models start by incorporating the full abstract as an initial summary and continue by adding sentences to it. For comparison purposes, we also show the results of BertSum (Liu and Lapata 2019) in both extractive and abstractive settings in the last block of the table.[8] We also present the

[8] Those results were obtained by running the publicly released source codes on the same data.

performance on the 2019 CL-SciSumm test dataset in the "Appendix 2" as a point of reference (see footnote 6).

We can see that models with linear layers outperform other models in both scenarios. More importantly, we note that linear models that use only the cited text spans and the abstract of the paper obtained the best ROUGE-2 score while maintaining comparable performance on ROUGE-1 and ROUGE-L scores. We thus show that the combination of abstracts and cited text spans is a valid substitute to using the full text and can simplify the summarisation task when it comes to summarising long scientific documents.

Based on our observations, most of the summary sentences are selected from the beginning of the input document. Indeed, when we calculated the ROUGE scores for each abstract on its own, we obtain the best ROUGE-2 score (25.54), although the ROUGE-1 and ROUGE-L scores are lower than those in our proposed methods. This result may be explained by the fact that the abstract aims to communicate the main ideas described in the paper.

For models that selected sentences from a combination of abstracts and citances, augmenting them with the abstract yields lower ROUGE scores although the difference is not significant. The same behaviour can be observed in the case of selecting sentences from full texts, i.e., augmenting abstracts does not help to improve the performance. However, in these cases, the differences in ROUGE scores are bigger, indicating that augmenting abstracts possibly adds more noise to the models. This aligns with our observations on the CL-SciSumm dataset, which showed that only 15.98% of the sentences are selected/modified from the abstract while the majority of them (68.95%) are from the body of the paper.

Models that select sentences from abstracts and citances achieved the highest ROUGE-2 scores among all of the models. In interpreting these results, we need to keep in mind that the training data—ScisummNet was created in a similar way, i.e., the human annotators only read abstracts and cited text spans of papers in order to produce their summaries.

Conclusions

We have presented approaches to identify cited text spans and generate scientific summaries that build on pre-trained encoders focusing mainly on BERT-based models. For both the tasks of cited text span identification and scientific summarisation we examined the potential for adapting architectures that have proved to be efficient in the generic and newswire domain and explored methods for adapting them to scientific publications. We compared our methods on a range of different architectures, training configurations and input variations to assess their robustness and potential, especially when applied to small annotated datasets, such as the one provided for cited-text span identification in the CL-SciSumm shared tasks.

For the cited text spans, we have based our experiments on BERT and compared with other architectures, as well as domain fine-tuning approaches. Overall, using BERT-based architectures outperformed both BiMPM and XLNET, which have both been shown to perform well in sentence similarity and inference tasks. We attribute this to the fact that the BERT language model was trained using the next sentence prediction task alongside label masking. In terms of fine-tuning, the use of additional domain specific dataset for fine-tuning prior to using the manually annotated data helped to improve the performance. It appears that while fine-tuning on large, unlabelled data is better in terms of strict F-score performance, fine-tuning on the automatically augmented task-specific data produces better performance when approximate targets are considered. Since the use of approximate targets can still produce meaningful results, we intend to further pursue the direction of augmenting data, focusing more on distant learning and instance weighting techniques, which could help to improve performance in this task.

We have demonstrated the suitability of cited text spans as a replacement for the full text of a publication, when used as input to a BERT-based classifier for scientific summarisation. We have showed that indeed, such an approach can reduce the input size (and thus time) needed to generate scientific summaries, while maintaining and even further improving efficiency (by ROUGE-2 metric).

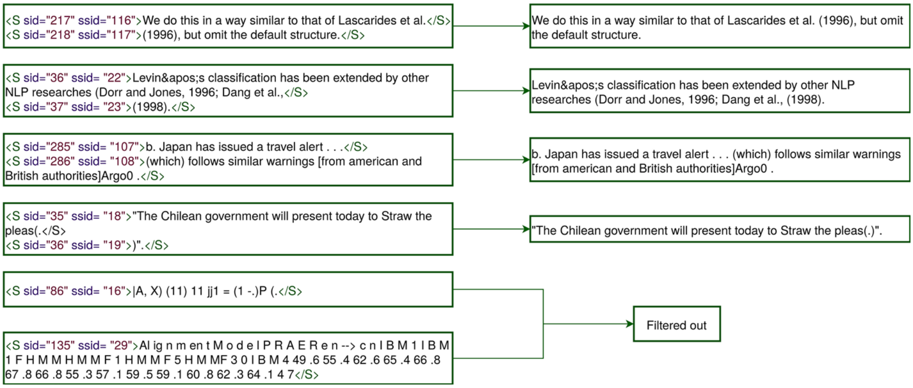
We also compared different summarisation layers used on top of the BERT-based summarisation model and found that there is no statistically significant improvement when using BiLSTM or transformer layers, in terms of the obtained F-score performance. We hence use a fully connected linear summarisation layer. Based on the experimental results a fully connected summarisation layer that selects sentences from the abstract and the cited text spans is the optimal configuration and obtains comparable (and even better) ROUGE scores to those obtained for newswire articles.

Acknowledgements This work was partly supported by the EPSRC Doctoral Prize award [EP/N509565/1]; the HSE Discovering Safety, Lloyd’s Register Foundation; and the Thomas Ashton Institute.

Appendix 1: OCR pre-processing

As mentioned in “Data resources and pre-processing” section both the 2018 dataset, the 2019 augmented dataset, and the ACL anthology corpus are derived using OCR methods. As such we observed several errors both in terms of sentence splitting and word tokenisation. Our pre-processing efforts aimed to correct some of the observed error patterns, focusing mostly on the sentence splitting errors. We present the main pre-processing rules below, and in Figs. 4 and 5; the full pre-processing code is made available as a Python notebook.

All CL-SciSumm datasets were processed during the training instance generation to remove the following errors:



```
<S sid="217" ssid="116">We do this in a way similar to that of Lascarides et al.</S>
<S sid="218" ssid="117">(1996), but omit the default structure.</S>
```

We do this in a way similar to that of Lascarides et al. (1996), but omit the default structure.

```
<S sid="36" ssid= "22">Levin&apos;s classification has been extended by other
NLP researches (Dorr and Jones, 1996; Dang et al.,</S>
<S sid="37" ssid= "23">(1998).</S>
```

Levin&apos;s classification has been extended by other NLP researches (Dorr and Jones, 1996; Dang et al., (1998).

```
<S sid="285" ssid= "107">b. Japan has issued a travel alert . . .</S>
<S sid="286" ssid= "108">(which) follows similar warnings [from american and
British authorities]Argo0 .</S>
```

b. Japan has issued a travel alert . . . (which) follows similar warnings [from american and British authorities]Argo0 .

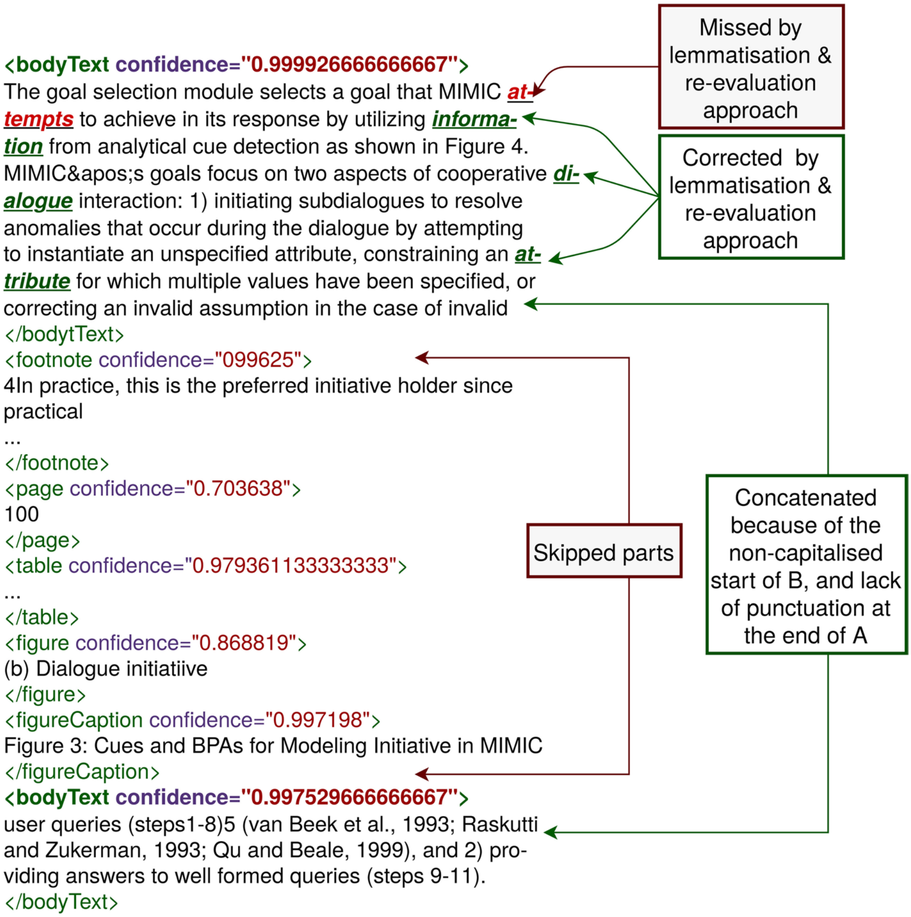
```
<S sid="35" ssid= "18">"The Chilean government will present today to Straw the
pleas(.</S>
<S sid="36" ssid= "19">)".</S>
```

"The Chilean government will present today to Straw the pleas(.)".

```
<S sid="86" ssid= "16">|A, X) (11) 11 jj1 = (1 -.)P (.</S>
```

```
<S sid="135" ssid= "29">Al ig n m en t M o d e l P R A E R e n --> c n I B M 1 I B M
1 F H M M H M M F 1 H M M F 5 H M MF 3 0 I B M 4 49 .6 55 .4 62 .6 65 .4 66 .8
67 .8 66 .8 55 .3 57 .1 59 .5 59 .1 60 .8 62 .3 64 .1 4 7</S>
```

Fig. 4 Examples of the pre-processing outputs for the CL-SciSumm dataset



```
<bodyText confidence="0.999926666666667">
The goal selection module selects a goal that MIMIC at-
tempts to achieve in its response by utilizing informa-
tion from analytical cue detection as shown in Figure 4.
MIMIC&apos;s goals focus on two aspects of cooperative di-
alogue interaction: 1) initiating subdialogues to resolve
anomalies that occur during the dialogue by attempting
to instantiate an unspecified attribute, constraining an at-
tribute for which multiple values have been specified, or
correcting an invalid assumption in the case of invalid
</bodytText>
<footnote confidence="099625">
4In practice, this is the preferred initiative holder since
practical
...
</footnote>
<page confidence="0.703638">
100
</page>
<table confidence="0.979361133333333">
...
</table>
<figure confidence="0.868819">
(b) Dialogue initiatiive
</figure>
<figureCaption confidence="0.997198">
Figure 3: Cues and BPAs for Modeling Initiative in MIMIC
</figureCaption>
<bodyText confidence="0.997529666666667">
user queries (steps1-8)5 (van Beek et al., 1993; Raskutti
and Zukerman, 1993; Qu and Beale, 1999), and 2) pro-
viding answers to well formed queries (steps 9-11).
</bodyText>
```

Missed by lemmatisation & re-evaluation approach
Corrected by lemmatisation & re-evaluation approach
Skipped parts
Concatenated because of the non-capitalised start of B, and lack of punctuation at the end of A

Fig. 5 Example of the processing approach for the ACL-ARC XML output. The high-lighted parts are the ones used for the fine-tuning data

- *Erroneous splitting at periods* (".") The sentence splitting appeared to consistently split sentences where a period symbol was followed by a non alphabetical character.
- *Erroneous splitting after comma* (",") Refers to erroneous splitting after commas, where the next sentence did not start with a capital letter.
- *Erroneous splitting within parentheses* ("(,)") Refers to erroneous splitting when there is an extra parentheses left bracket and in the following sentence there is a corresponding (unpaired) right bracket.
- *Erroneous splitting after* "..." *or* "...".

For the generation of negative instances, sentences were further filtered to remove noisy sentences. The percentages in the filters presented below were determined by testing against a small sample of 5 documents from the CL-SciSumm corpus, that were manually inspected for sentence validity.

Filtered sentences fulfil one or more of the properties below:

- Contain more than 30% of tokens whose lemmas do not correspond to known words (compare against WordNet lexicon Miller 1995).
- Contain more than 20% of alphabetical single characters.
- Contain less than 10% of tokens whose lemmas correspond to known words (compare against WordNet lexicon Miller 1995).

Examples of sentences filtered/corrected in prepossessing are presented in Fig. 4.

Additional corrections and filtering measures for the ACL anthology corpus (see also Fig. 5) involved filtering for sections with text and restoring erroneously split parts of text because of page or column breaks.

- Select only the < *bodytext* > elements. This effectively means ignoring footnotes, figures, tables, page numbers etc.
- Apply nltk sentence splitter,[9] to properly split larger chunks of text.
- If the first word of a "bodytext" element is lower-cased, check the last sentence of the previous bodytext. If it does not end with a punctuation mark, concatenate the two.
- Lemmatise text and verify whether the lemma of the first word of each chunk is a valid one (compare against WordNet lexicon Miller 1995). If not, and if the previous sentence does not end with a punctuation mark or ends with a dash ("-"), concatenate the sentences and reconsider the lemma validity.
- Lemmatise text and verify whether the lemma of the first word of each chunk is a valid one (compare against WordNet lexicon Miller 1995). If not, and if the previous word does not ends with a dash ("-"), remove the dash, concatenate tokens and reconsider the lemma validity.
- Maintain only parsed documents with confidence > 0.6 based on the provided confidence measures.

[9] https://www.nltk.org/api/nltk.tokenize.html.

Appendix 2: Performance on 2019 test data for the CL-SciSumm challenge

For Task 1A (cited text span identification) we submitted 11 runs. We can see that similarly to our experiments in "Fine-tuning on augmented data" section the ACL model seems to outperform other approaches. However, with the exception of the BiMPM model (run 10, 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).

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.

Table 9 Submitted system and obtained performance for each run in Task 1A (cited text span identification)

Run	System	Sent. Ov. (F1)	R-SU4 (F1)
1	BERT	0.093	0.06
2	ACL	**0.126**	0.075
3	BERT 19-AUG	0.097	0.062
4	BERT 19-AUG-FT	0.11	0.062
5	BERT 19-AUG-FT [OV]	0.12	0.072
6	ACL 19-AUG-FT	0.118	**0.079**
7	CNN+SciBERT	0.078	0.048
8	BiMPM 2019 [OV]	0.074	0.051
9	BiMPM 19-AUG	0.012	0.018
10	BiMPM 2019 [OV] top-2	0.11	0.073
11	BERT top-2	0.062	0.052

Bold values correspond to the highest obtained performance for each metric

Table 10 Submitted system and obtained performance in Task 2 (scientific summarisation)

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

Bold values correspond to the highest obtained performance for each metric

Appendix 3: Feature-based approach with CNN: initial experiments

In this section we present the output of a set of experiments performed with the CNN + BERT/SciBERT embedding configurations in order to determine the optimal among the configurations to use for further experiments. The experiments focus on two different hypotheses:

1. Examining whether the use of additional features capturing the position of the citing and cited sentences would help the performance. The underlying assumption here is that passages in certain sections, such as introduction or conclusions might be cited more often, or that citing sentences found in the methods of the CP are more likely to cite sentences from the corresponding section in the RP. For this set of experiments we encode the sentence ids (sid) and the section ids (ssid) for each sentence and concatenate them as features in the linear layer of the CNN. We compare adding the sentence ids to using only the BERT features (PLAIN) in Table 11.
2. Examining whether the use of [CLS], [CLS]+[SEP], or the full BERT embedding vector is more beneficial as input features for the CNN configuration. We have seen that CLS captures the core information about the relation between the two sentences which is the information of interest in this task, hence we assume that focusing only on the sentence instead of token embeddings might be more beneficial, while it also reduces the dimensionality of the feature vectors. The results of the evaluation using only the [CLS] and [CLS] + [SEP] embeddings can be observed in Table 12.

We note that these experiments were evaluated on the development set (not used during training) consisting of 8 documents of the CL-SciSumm 2019 dataset. 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.

We conclude that the addition of SID and SSID features does not contribute to the performance. Moreover, we notice that using the combination of [CLS] and [SEP] embeddings as input is better than using full sentence embeddings for both systems. Hence we use this combination for the tenfold cross validation experiments presented in the main manuscript.

Table 11 Results for plain embedding concatenation, feature based approach for BERT and SciBERT

Model	Extra features	Recall	Precision	F-score
BERT	SID + SSID	0.187	0.095	0.126
	SID	0.197	0.107	0.139
	PLAIN	**0.235**	**0.124**	**0.163**
SciBERT	SID + SSID	0.167	0.085	0.113
	SID	0.183	0.096	0.126
	PLAIN	0.203	0.104	0.137

Bold values correspond to the highest obtained performance for each metric

Table 12 Results for ablation tests with variations on embeddings and extra feature combination

Model	Embeddings	Extra features	Recall	Precision	F-score
BERT	[CLS]	SID + SSID	0.159	0.102	0.124
		SID	0.212	0.106	0.142
		PLAIN	0.254	0.131	0.173
	[CLS]+[SEP]	SID + SSID	0.162	0.103	0.126
		SID	0.209	0.104	0.139
		PLAIN	**0.260**	**0.134**	**0.177**
SciBERT	[CLS]	SID + SSID	0.190	0.095	0.127
		SID	0.190	0.096	0.127
		PLAIN	0.244	0.126	0.166
	[CLS]+[SEP]	SID + SSID	0.183	0.092	0.123
		SID	0.186	0.095	0.126
		PLAIN	0.251	0.129	0.170

Bold values correspond to the highest obtained performance for each metric

References

Abu-Jbara, A., & Radev, D. (2011). 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.

Abura'ed, A., Bravo, A., Chiruzzo, L., & Saggion, H. (2018). Lastus/taln+ inco@ cl-scisumm 2018-using regression and convolutions for cross-document semantic linking and summarization of scholarly literature. In *Proceedings of the 3rd joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL2018)*. Ann Arbor, Michigan (July 2018).

Aggarwal, P., & Sharma, R. (2016). Lexical and syntactic cues to identify reference scope of citance. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)* (pp. 103–112).

Agrawal, K., & Mittal, A. (2018) Iiit-h@ clscisumm-18. In *BIRNDL@ SIGIR* (pp. 130–133).

Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., Dunkelberger, J., Elgohary, A., Feldman, S., & Ha, V., et al. (2018). Construction of the literature graph in semantic scholar. arXiv:1805.02262.

Baruah, G., & Kolla, M. (2018). Klick labs at cl-scisumm 2018. In *BIRNDL@ SIGIR* (pp. 134–141).

Beltagy, I., Cohan, A., & Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text. arXiv:1903.10676.

Bird, S., Dale, R., Dorr, B.J., Gibson, B., Joseph, M.T., Kan, M.Y., Lee, D., Powley, B., Radev, D.R., & Tan, Y.F. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics.

Bornmann, L., & Daniel, H. D. (2009). The state of h index research. *EMBO Reports*, *10*(1), 2–6.

Cao, Z., Li, W., & Wu, D. (2016). Polyu at cl-scisumm 2016. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)* (pp. 132–138).

Carbonell, J., & Goldstein, J. (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 335–336).

Chandrasekaran, M. K., Yasunaga, M., Radev, D., Freitag, D., Kan, M. Y., & (2019). Overview and Results: CL-SciSumm SharedTask 2019. In *Proceedings of the 4th joint workshop on bibliometric-enhancedinformation retrieval and natural language processing for digital libraries (BIRNDL 2019) @ SIGIR 2019*. Paris.

Chang, L. L. H., Phoa, F. K. H., & Nakano, J. (2019). A new metric for the analysis of the scientific article citation network. *IEEE Access*, *7*, 132027–132032.

Cohan, A., & Goharian, N. (2017). Scientific article summarization using citation-context and article's discourse structure. arXiv:1704.06619.

Conroy, J. M., & O'leary, D. P. (2001). Text summarization via hidden Markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 406–407). ACM.

Conroy, J. M., Schlesinger, J. D., & O'Leary, D. P. (2006). Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on main conference poster sessions* (pp. 152–159). Association for Computational Linguistics.

Contractor, D., Guo, Y., & Korhonen, A. (2012). Using argumentative zones for extractive summarization of scientific articles. *Proceedings of COLING*, *2012*, 663–678.

Councill, I. G., Giles, C. L., & Kan, M. Y. (2008). Parscit: An open-source crf reference string parsing package. *LREC*, *8*, 661–667.

Craswell, N. (2009). Mean reciprocal rank. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of database systems*. Boston, MA: Springer.

De Moraes, L. F., Das, A., Karimi, S., & Verma, R. M. (2018). University of houston@ cl-scisumm 2018. In *BIRNDL@ SIGIR* (pp. 142–149).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423.

Ding, Y., Rousseau, R., & Wolfram, D. (2016). *Measuring scholarly impact*. Berlin: Springer.

Duan, C., Cui, L., Chen, X., Wei, F., Zhu, C., & Zhao, T. (2018). Attention-fused deep matching network for natural language inference. In *IJCAI* (pp. 4033–4040).

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, *16*(2), 264–285.

Fergadis, A., Pappas, D., & Papageorgiou, H. (2019). Athena@ cl-scisumm 2019: Siamese recurrent bi-directional neural network for identifying cited text spans.

Fister, I, Jr., Fister, I., & Perc, M. (2016). Toward the discovery of citation cartels in citation networks. *Frontiers in Physics*, *4*, 49.

Galgani, F., Compton, P., & Hoffmann, A. (2012). Citation based summarisation of legal texts. In *Pacific Rim International Conference on Artificial Intelligence* (pp. 40–52). Springer.

Galgani, F., Compton, P., & Hoffmann, A. (2015). Summarization based on bi-directional citation analysis. *Information Processing and Management*, *51*(1), 1–24.

Hassan, S. U., Imran, M., Iqbal, S., Aljohani, N. R., & Nawaz, R. (2018). Deep context of citations using machine-learning models in scholarly full-text articles. *Scientometrics*, *117*(3), 1645–1662.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv:1801.06146.

Hutchins, B. I., Yuan, X., Anderson, J. M., & Santangelo, G. M. (2016). Relative citation ratio (RCR): A new metric that uses citation rates to measure influence at the article level. *PLoS biology*, *14*(9), e1002541.

Jaidka, K., Chandrasekaran, M. K., Jain, D., & Kan, M. Y. (2017). The CL-SciSumm shared task 2017: Results and key insights. In *BIRNDL@ SIGIR (2)* (pp. 1–15).

Jaidka, K., Chandrasekaran, M. K., Jain, D., & Kan, M. Y. (2018). The CL-SciSumm shared task 2018: Results and key insights. In *BIRNDL@ SIGIR (2)* (pp. 1–15).

Jaidka, K., Chandrasekaran, M. K., Rustagi, S., & Kan, M. Y. (2016). 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).

Jha, S., Chaurasia, A., Sudhakar, A., & Singh, A. K. (2017). Reference scope identification for citances using convolutional neural networks. In *Proceedings of the 14th international conference on natural language processing (ICON-2017)* (pp. 23–32).

Karimi, S., Moraes, L. F., Das, A., & Verma, R. M. (2017). University of houston@ CL-SciSumm 2017: Positional language models, structural correspondence learning and textual entailment. In *BIRNDL@ SIGIR (2)* (pp. 73–85).

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. arXiv:1909.05858.

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv:1408.5882.

Klampfl, S., Rexha, A., & Kern, R. (2016). Identifying referenced text in scientific publications by summarisation and classification techniques. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)* (pp. 122–131).

Kupiec, J., Pedersen, J., & Chen, F. (1999). A trainable document summarizer. In *Advances in automatic summarization* (pp. 55–60).

La Quatra, M., Cagliero, L., & Baralis, E. (2019). Poli2sum@ cl-scisumm-19: Identify, classify, and summarize cited text spans by means of ensembles of supervised models.

Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017). Race: Large-scale reading comprehension dataset from examinations. arXiv:1704.04683.

Li, L., Chi, J., Chen, M., Huang, Z., Zhu, Y., & Fu, X. (2018). Cist@ clscisumm-18: Methods for computational linguistics scientific citation linkage, facet classification and summarization. In *BIRNDL@ SIGIR* (pp. 84–95).

Li, L., Zhang, Y., Mao, L., Chi, J., Chen, M., & Huang, Z. (2017). Cist@ clscisumm-17: Multiple features based citation linkage, classification and summarization. In *BIRNDL@ SIGIR (2)* (pp. 43–54).

Li, L., Zhu, Y., Xie, Y., Huang, Z., Liu, W., Li, X., & Liu, Y. (2019). Cist@ clscisumm-19: Automatic scientific paper summarization with citances and facets. In *BIRNDL2019*.

Lin, C.Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Barcelona: Association for Computational Linguistics. https://www.aclweb.org/anthology/W04-1013.

Lin, C. Y., & Hovy, E. (2000). The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on computational linguistics-volume 1* (pp. 495–501). Association for Computational Linguistics.

Liu, Y. (2019). Fine-tune BERT for extractive summarization. arXiv:1903.10318 [cs].

Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders.

Liu, Y., Titov, I., & Lapata, M. (2019). Single document summarization as tree induction. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (Long and Short Papers)* (pp. 1745–1755).

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, *2*(2), 159–165.

Ma, S., Zhang, H., Xu, J., & Zhang, C. (2018). Njust@ clscisumm-18. In *BIRNDL@ SIGIR* (pp. 114–129).

Ma, S., Zhang, H., Xu, T., Xu, J., Hu, S., & Zhang, C. (2019). Ir&tm-njust@ clscisumm-19. In *BIRNDL2019*.

Mihalcea, R., Corley, C., Strapparava, C., et al. (2006). Corpus-based and knowledge-based measures of text semantic similarity. *Aaai*, *6*, 775–780.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures.

Miller, G. A. (1995). Wordnet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishnan, P., Qazvinian, V., Radev, D., & Zajic, D. (2009). Using citations to generate surveys of scientific paradigms. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics* (pp. 584–592).

Mollá, D., Jones, C., & Sarker, A. (2014). Impact of citing papers for summarisation of clinical documents. *Proceedings of the Australasian Language Technology Association Workshop*, *2014*, 79–87.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Neculoiu, P., Versteegh, M., & Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st workshop on representation learning for NLP* (pp. 148–157).

Nicosia, M., & Moschitti, A. (2017). Accurate sentence matching with hybrid siamese networks. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 2235–2238). ACM.

Nie, Y., & Bansal, M. (2017). Shortcut-stacked sentence encoders for multi-domain inference. arXiv:1708.02312

Nomoto, T. (2016). Neal: A neurally enhanced approach to linking citation and reference. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)* (pp. 168–174).

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

Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014)* (pp. 1532–1543).

Prasad, A. (2017). Wing-nus at cl-scisumm 2017: Learning from syntactic and semantic similarity for citation contextualization. In *BIRNDL@SIGIR(2)* (pp. 26–32).

Qazvinian, V., & Radev, D.R. (2008). 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.

Radev, D. R., Muthukrishnan, P., Qazvinian, V., & Abu-Jbara, A. (2013). The ACL anthology network corpus. *Language Resources and Evaluation*, *47*(4), 919–944.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. arXiv:1606.05250.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 379–389). Lisbon: Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1044. https://www.aclweb.org/anthology/D15-1044.

Saggion, H., AbuRa'ed, A., & Ronzano, F. (2016). Trainable citation-enhanced summarization of scientific articles. In G. Cabanac, M. K. Chandrasekaran, I. Frommholz, K. Jaidka, M. Kan, P. Mayr, D. Wolfram (Eds.), *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL); [place unknown]: CEUR Workshop Proceedings 2016* (pp. 175–186). CEUR Workshop Proceedings. 2016 June 23; Newark, United States.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th annual meeting of the association for computational linguistics* volume 1: Long Papers (pp. 1073–1083). Association for Computational Linguistics, Vancouver, Canada. https://doi.org/10.18653/v1/P17-1099. https://www.aclweb.org/anthology/P17-1099.

Shen, D., Sun, J. T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using conditional random fields. *IJCAI*, *7*, 2862–2867.

Teufel, S., Siddharthan, A., & Tidhar, D. (2006). 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.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., & Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1386–1393).

Wang, P., Li, S., Wang, T., Zhou, H., & Tang, J. (2018). Nudt@ clscisumm-18. In *BIRNDL@ SIGIR* (pp. 102–113).

Wang, Z., Hamza, W., & Florian, R. (2017). 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). https://doi.org/10.24963/ijcai.2017/579.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., & Funtowicz, M., et al. (2019). Transformers: State-of-the-art natural language processing. arXiv:1910.03771.

Xu, H., Martin, E., & Mahidadia, A. (2015). Extractive summarisation based on keyword profile and language model. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 123–132).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5754–5764).

Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A. R., Li, I., Friedman, D., et al. (2019). Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 7386–7393.

Yeh, J. Y., Hsu, T. Y., Tsai, C. J., & Cheng, P. C. (2017). Reference scope identification for citances by classification with text similarity measures. In *Proceedings of the 6th international conference on software and computer applications* (pp. 87–91). ACM.

Zerva, C., Nghiem, M. Q., Nguyen, N. T., & Ananiadou, S. (2019). Nactem-uom@ cl-scisumm 2019. In *BiRNDL@SIGIR2019*.

Zhang, X., Wei, F., & Zhou, M. (2019). HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5059–5069). Association for Computational Linguistics, Florence, Italy https://doi.org/10.18653/v1/P19-1499. https://www.aclweb.org/anthology/P19-1499.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T. (2018). Neural document summarization by jointly learning to score and select sentences. arXiv:1807.02305.