CitRet: A Hybrid Model for Cited Text Span Retrieval

Amit Pandey[†*], **Avani Gupta**[*] **and Vikram Pudi**[†]
[†]Data Sciences and Analytics Center, Kohli Center on Intelligent Systems
International Institute of Information Technology, Hyderabad, India
{amit.pandey, avani.gupta}@research.iiit.ac.in
vikram@iiit.ac.in

Abstract

The paper aims to identify cited text spans in the reference paper related to the given citance in the citing paper. We refer to it as cited text span retrieval (CTSR). Most current methods attempt this task by relying on pre-trained off-the-shelf deep learning models like SciBERT. Though these models are pre-trained on large datasets, they underperform in out-of-domain settings. We introduce CitRet, a novel hybrid model for CTSR that leverages unique semantic and syntactic structural characteristics of scientific documents. This enables us to use significantly less data for finetuning. We use only 1040 documents for finetuning. Our model augments mildly-trained SBERT-based contextual embeddings with pre-trained non-contextual Word2Vec embeddings to calculate semantic textual similarity. We demonstrate the performance of our model on the CLSciSumm shared tasks. It improves the state-of-the-art results by over 15% on the F1 score evaluation.

1 Introduction

Citations are an integral part of scientific literature as they help better understand the relationships between scientific documents. Authors cite other papers to acknowledge their contributions, compare to their work, criticize, and improve upon their work. Citances often focus on the most important components of a scientific document. Moreover, citance-based summarization is also a widely studied field because it covers some insights that might not be present in abstract-based summarization (Elkiss et al., 2008).

However, a citance depends on the intention and opinion of the citing author and can be affected by epistemic value drift[1] (Cohan et al., 2015). Also, a citance in itself lacks sufficient details to capture the exact content of the referenced paper. Hence, identifying the correct context of the cited text can enable us to verify the biases (Zerva et al., 2020), overcome epistemic value drift, build dense knowledge graphs, and generate better summaries (Jaidka et al., 2019; Chandrasekaran et al., 2020). Furthermore, it also helps in qualitative analysis of the citations (Teufel et al., 2006). Motivated by these, research tasks and tracks such as BiomedSumm[2] and CLSciSumm lay significant emphasis on this fundamental and challenging problem of finding the exact cited text span. We refer to this task as cited text span retrieval (CTSR).

Most of the current methods targeting this problem are centered around fine-tuning deep neural networks. In this regard, transformer (Vaswani et al., 2017) based encoders such as BERT (Devlin et al., 2018) and SciBERT (Beltagy et al., 2019) have proven to be very effective and have outperformed standard baselines like LDA and TF-IDF. However, a major drawback of these methods is that they require large domain-specific datasets, often exceeding 1 million documents, to fine-tune.

This paper proposes CitRet, a hybrid CTSR model that performs well even in low-resourced domain-specific settings. We model the problem as a semantic textual similarity (STS) task. We exploit the distinctive semantic and syntactic structural characteristics of scientific literature, i.e., when a paper is cited, the cited text of the reference paper is often paraphrased in such a way that it still expresses the same central idea while also preserving certain keywords. Hence, we use these keywords, which are common to both the citance and the cited sentence, to find weighted contextual embeddings for the sentences. To find these weighted contextual embeddings, we use Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) fine-tuned to minimize cosine similarity loss on training data. However, when the training data is scarce, these contextual embeddings fail to cap-

[*]Equal contribution

[1]An example of epistemic value drift is citing a claim as a fact.

[2]http://www.nist.gov/tac/2014/BiomedSumm/

Proceedings of the 29th International Conference on Computational Linguistics, pages 4528–4536
October 12–17, 2022.

ture out-of-domain knowledge. To overcome this, we further leverage pre-trained non-contextual embeddings like Word2Vec (Mikolov et al., 2013) to capture the general domain knowledge. We use Word Mover's Distance (WMD) (Kusner et al., 2015) to find (dis)similarity scores based on these non-contextual embeddings. This hybrid approach of utilizing contextual and non-contextual embeddings enables CitRet to generalize well over unseen datasets. Definitions of the terms used throughout the paper are:

Reference paper (RP): A scientific document of which one or more sentences have been cited by another paper(s). **Citing paper (CP)**: A document that contains one or multiple citations to an RP. **Citance**: A sentence in CP that contains the reference to the RP. **Cited sentence**: The exact piece of the text belonging to the RP that a citance refers to. **Cited text span**: Span of the cited sentence(s) belonging to the RP corresponding to a citance.

The major contributions of this work are: 1) Proposing a simple yet effective CTSR model that requires less data for fine-tuning and is computationally inexpensive. We train only on the CL-SciSumm training dataset that consists of 40 manually annotated articles and 1000 automatically annotated articles. 2) Advancing the state-of-the-art (SOTA) to identify cited text span by over 15%. 3) Empirically validating the advantage of using the semantic and syntactic structure for CTSR.

2 Related Work

The task of CTSR requires modeling the relationship (similarity) between a citing and a candidate cited sentence. Early systems proposed using features based on TF–IDF (Yeh et al., 2017; Cao et al., 2016; Prasad, 2017) and n-grams or sentence graph overlap (Aggarwal and Sharma, 2016; Klampfl et al., 2016) in order to calculate similarity scores between the citing sentence and candidate sentences. Similarity measures such as Jaccard similarity and cosine similarity were commonly used to solve this task. (Bravo et al., 2018; Debnath et al., 2018; Kim and Ou, 2019; Pitarch et al., 2019). The problem has also been posed as a binary classification problem in Davoodi et al. (2018); Yeh et al. (2017); Zerva et al. (2020). In addition to traditional features such as TF-IDF and n-grams, prior methods have also proposed using learned distributed vector space representation (word embeddings) based features since they contain the semantic similarity information at the word level. Models using both non-contextual embeddings such as Word2Vec and contextual embedding methods like BERT have been utilized to find these word embeddings. These extracted features are further used as an input to machine learning algorithms like SVM (Ma et al., 2018), random forests (Wang et al., 2018), Word Mover's Distance (Li et al., 2018), CNN (Li et al., 2019; AbuRa'ed et al., 2018) or XGBoost (Syed et al., 2019; Pitarch et al., 2019). Furthermore, many approaches even adopted voting mechanisms and ensemble techniques on top of their models to improve their metrics (Chai et al., 2020; Wang et al., 2018; Ma et al., 2018, 2019; Quatra et al., 2019). The current best performing models exploit transformers fine-tuned on very large datasets (Chai et al., 2020; Zerva et al., 2019). Chai et al. (2020) also experimented with adding document level features to the model using special tokens. Other noteworthy approaches, like Aumiller et al. (2020) formulated the task as a search problem and used a two-step approach for retrieving relevant sentences for a given citation. They first find candidate sentences using Apache Solr and BM25 and then re-rank the retrieved sentences using a computationally expensive BERT-based re-ranker.

CTSR as Semantic Textual Similarity: We model the problem as a semantic textual similarity (STS) task. To this end, learning sentence embeddings, instead of word embeddings, has shown promise and improvement in performance (Reimers and Gurevych, 2019). Using pooling strategies such as mean or max pooling of word embeddings has proven to be an efficient way of obtaining sentence embeddings. SBERT (Reimers and Gurevych, 2019) by default uses mean pooling. Chen et al. (2018) further explored generalized pooling strategies to enhance sentence embeddings. CNN-based models have also been used to encode sentences into fixed length vectors (Jiao et al., 2018). To improve performance on sentence matching tasks, Liu et al. (2020) proposed syntax- and semantics-aware BERT(SS-BERT), which implicitly integrates syntactic and semantic information of sentences. Unnam et al. (2022) showed that sentence embeddings could be further improved by employing principal component removal based denoising as a post-processing step.

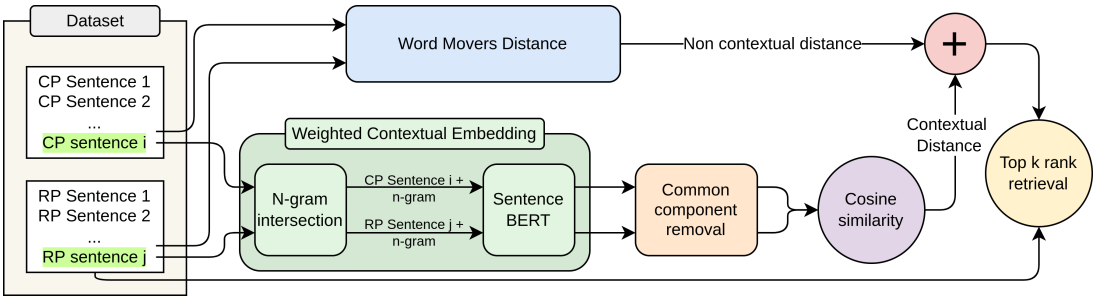

Figure 1: Illustration of the CitRet model. WMD and weighted contextual embeddings (WCEs) are calculated for an input pair. The WCEs are then denoised using the common component removal technique. These denoised WCEs are used to find cosine similarity between the sentences of the input pair. Finally, WMD and cosine scores are added, and top k similar sentences in an RP for a citance are retrieved.

3 Methodology

We formulate this task of CTSR as finding semantic textual similarity between a citance and all the sentences of an RP, i.e., to find the cited text span for a given citance, we pick the top k similar sentences in the RP. We refer to a <citance, a sentence in the RP> pair as an *input pair*. As shown in Figure 1, an input pair is first pre-processed by lowercasing the tokens, removing the stop words, and removing the special characters. Then to find the final similarity scores, CitRet employs a mix of cosine scores using weighted contextual embeddings (contextual distance) and Word Mover's Distance scores (non-contextual distance) using pre-trained non-contextual embeddings. Now, we explain each component of the pipeline in detail.

3.1 Contextual Distance

Contextual distance between the sentences is calculated using contextual sentence embeddings. The proposed model uses finetuned SBERT to learn these contextual embeddings for an input pair. SBERT returns a fixed-length dense vector for an input sentence (sentence embedding), irrespective of the length of the input sentence[3]. To yield the final sentence embeddings, CitRet follows three steps: 1) Finetuning the SBERT, 2) Finding the weighted contextual embeddings for each sentence pair, and 3) Denoising the embeddings.

3.1.1 Finetuning the SBERT

To finetune SBERT siamese networks, we use cosine similarity loss. As training examples, we pass sentence pairs annotated with cosine similarity scores on a scale of 0 to 1. For each citance, we pass 5 sentence pairs of 3 different types, i.e., one pair with the actual cited text having a similarity score of 1, two pairs with randomly selected sentences from other RP having a similarity score of 0, and two pairs with randomly selected sentences belonging to the same RP having similarity score of 0.3. This helps us model relations between the sentences of the same documents and sentences of different documents.

3.1.2 Weighted contextual embeddings

When an RP is cited, the information that can be extracted from a citance about the RP depends upon the intention, and the opinion of the citing author(s) (Zerva et al., 2020). However, when the cited sentences are referred to, some key ideas and keywords are preserved, as depicted in Figure 2. CitRet exploits this characteristic of the scientific documents to find weighted contextual embeddings for the input pair.

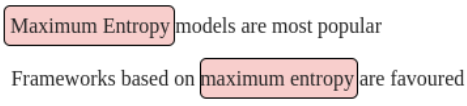

Figure 2: n-gram intersection of two sentences

SBERT takes the mean of all the word embeddings to calculate the sentence embedding. After fine-tuning SBERT on domain-specific data, it is able to learn contextual embedding for a sentence. To leverage this contextual learning capability of SBERT and to find weighted contextual embeddings (WCEs) for the sentences, we use a very simple and intuitive strategy of concatenating the common keywords to the input pair before passing it to the SBERT (we concatenate the keyword to both the sentences of the pair). These keywords are extracted by finding common n-gram intersections between the sentences of the input pair. In the example shown in Figure 2, *maximum entropy* is the common keyword (bigram). Concatenating these

[3]Please refer to Appendix (A.1) for more details.

n-grams results in the common keywords having more weight in the sentence embeddings due to the mean pooling operation. Therefore, the sentence embedding vectors of the pair come closer in the dense vector space if they share some keywords. Here, number *n* can be optimized empirically, and in our tests, we get the best results for bigrams.

3.1.3 Denoising

We further modify the WCEs that we get from the previous step by using a denoising technique adapted from *piecewise common component removal* method proposed in Ethayarajh (2018). Here, the common components refer to the common topics (discourse themes) that exist throughout the document (RP and CP) and can be considered as noise. Thus, removing these common components can be understood as downgrading the unimportant components (common discourse) and focusing on the components that have more discriminatory power. This helps in denoising the embeddings (Arora et al., 2017). Since cosine-similarity treats all dimensions equally (Reimers and Gurevych, 2019), denoising becomes critical in making it more focused. Consequently, the cosine similarity scores calculated using denoised embeddings become more relevant (Arora et al., 2017).

$$\tilde{v} = v - \sum_{i}^{m} \lambda_i \, proj_{pc_i} \, v \quad \text{, where } \lambda_i = \frac{\sigma_i^2}{\sum_{j=1}^{m} \sigma_j^2}$$

These common discourse vectors are estimated as the principal components for a set of WCEs. These principal components are calculated by singular value decomposition of $A_{l \times d}$ matrix, where l is the number of sentences in the document (RP and CP), and d is the dimension of the WCEs. To get the final denoised sentence vector $\tilde{v}$, we subtract from the original sentence vector v, the weighted sum of the projections of the vector v on the first $m(=3)$ principal components $pc_{i..m}$. The projections $proj_{pc_i} \, v$ are weighted by λ_i, where λ_i is the proportion of variance σ_i (singular value) captured by the principal component pc_i.

3.2 Non-contextual Distance

CitRet uses both the supervised and unsupervised techniques to calculate the final similarity scores to generalize well over unseen datasets. It augments contextual distance calculated using mildly-trained SBERT with non-contextual distance calculated using unsupervised WMD technique[4]. Arora et al. (2017) and Reimers and Gurevych (2019) note that even simple techniques such as computing the average of pre-trained embeddings can outperform sophisticated techniques such as BERT in unsupervised textual similarity tasks. As discussed, a cited text is usually paraphrased around a keyword in such a way that it still expresses the same central idea. Since WMD uses the high-quality Word2Vec model embeddings having a vocabulary size of 3 million, it can capture knowledge related to these general domain words that fine-tuning a deep learning model with low training data might not be able to extract (Kusner et al., 2015). Figure 3 demonstrates WMD's ability to capture relations in the general domain setting. The arrows represent the flow between two words of an input pair. It may be observed how *models* flows to *frameworks* and *popular* to *favoured*. It can be noted that the words *popular* and *favoured* are general domain words (non-scientific terms) and might not appear very frequently in a scarce domain-specific dataset. Hence, the semantic relationships between these general domain words are better captured by WMD.

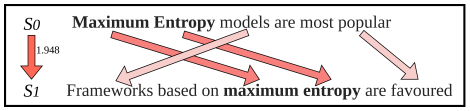

Figure 3: Flow between 2 sentences S_0 and S_1 using WMD

4 Experiments and Results

We demonstrate the performance of the proposed method on CL-SciSumm shared task (Jaidka et al., 2019; Chandrasekaran et al., 2019, 2020) task 1(a), where for each citance, we need to identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. These cited text spans range from the granularity of a sentence fragment to several consecutive sentences. We pick top $k = 3$ similar candidate cited sentences for a given citance. CitRet is trained only on the CL-SciSumm training dataset that consists of 40 manually annotated articles and 1000 low quality document sets that were automatically annotated using neural networks. We do not use any external corpora to fine-tune our model. We evaluate our model's performance against gold label annotations for the CL-SciSumm test set of 20 documents.

[4]Please refer to Appendix A.2 for a detailed explanation of WMD.

Method	Recall	Precision	F1
ACL 2018	-	-	0.126
BERT 2018/19 OV+2018FT	-	-	0.120
SciBERT 2018	-	-	0.078
SciBERT-SemBERT	**0.2459**	0.1318	0.1716
SciBer-ACLBERT	0.2265	0.1244	0.1606
SBERT†	0.1879	0.1023	0.1325
SBERT + WCE†	0.1815	0.1647	0.1727
Denoising (SBERT+WCE+D)†	0.1901	0.1724	0.1808
CitRet (SBERT+WCE+D+WMD)†	0.2080	**0.1888**	**0.1979**

Table 1: Performance comparison of our model with the baseline models. The last 4 rows show the ablation study of our model marked with †. D denotes denoising step.

We consider the SOTA models of 2019 and 2020 CL-SciSumm tasks as baselines. Table 1 shows that CitRet performs the best in quantitative metrics (F1 and Precision) and outperforms 2019 SOTA (*ACL 2018*) by over 57% and 2020 SOTA (*SciBERT-SemBERT*) by over 15% on F1 score evaluation. It can be noted that using just the *SBERT + WCE* component outperformed all the baseline SOTA models that use much larger datasets (exceeding 1 million) for finetuning[5]. This empirically validates that using the semantic and syntactic structure for CTSR can significantly improve the results. It should be noted that *Denoising* and *WMD* further improve the performance.

5 Discussion

As can be observed from Table 1, the proposed method significantly improves the F1 score (+15%) and Precision(+43%) with some loss in Recall(15%). Our approach focuses on Precision (a measure of the quality of retrieval) over Recall (a measure of quantity) because, for the given task, the probability of getting false positives is very high. Hence a higher precision results in a more concise and accurate summarization.

The proposed approach is in line with the recommendation made by the task organisers to exploit the structural and semantic characteristics that are unique to scientific documents to enrich the embeddings. The paper proposes a simple and computationally inexpensive alternative to the current state-of-art model in the form of CitRet. It leverages both contextual and non-contextual embeddings. CitRet also combines a supervised model and an unsupervised model. This hybrid architecture provides performance and robustness against noisy training samples. The components of the model are lightweight (do not require extensive fine-tuning), faster, explainable, and intuitive. This highlights how other statistical machine learning techniques can be leveraged along with modern deep neural network architectures to compensate for the lack of quality training data and outperform computationally expensive architectures.

It may also be noted that while our method beats the baselines by large margins and achieves a new SOTA, the absolute values are still rather low because of the non-triviality of the task. The task becomes particularly challenging because of the low-quality training data and subjectivity of the annotators. Hence, we believe that there is a scope for further improvement, and the problem demands greater exploration.

6 Conclusion

In this paper, we propose CitRet, a novel model for cited text span retrieval. CitRet outperforms the current SOTA models by significant margins (15% F1). The proposed model is quite simple, computationally inexpensive, improves generalization, and does not require any large external datasets to fine-tune. However, considering the non-triviality of the task, this paper proposes a new approach for further exploration of the task.

Acknowledgments

We thank IHub-Data, IIIT Hyderabad[6] for financial support. We thank Mr. Narendra Babu Unnam, DSAC, IIIT Hyderabad, for his comments that greatly improved the manuscript.

References

Ahmed Ghassan Tawfiq AbuRa'ed, Àlex Bravo Serrano, Luis Chiruzzo, and Horacio Saggion. 2018. Lastus/taln+ inco@ cl-scisumm 2018-using regression and convolutions for cross-document semantic linking and summarization of scholarly literature. In *Mayr P, Chandrasekaran MK, Jaidka K, editors. BIRNDL 2018. 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries; 2018 Jul 21; Ann Arbor, MI.[place unknown]: CEUR; 2018. p. 150-63.* CEUR Workshop Proceedings.

Peeyush Aggarwal and Richa Sharma. 2016. Lexical and syntactic cues to identify reference scope of citance. In *Proceedings of the joint workshop on bibliometric-enhanced information retrieval and*

[5]Please refer to Appendix B for details of the experimental setup of the baseline models and ablation study analysis.

[6]https://ihub-data.iiit.ac.in/

natural language processing for digital libraries (BIRNDL), pages 103–112.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Dennis Aumiller, Satya Almasian, Philip Hausner, and Michael Gertz. 2020. UniHD@CL-SciSumm 2020: Citation extraction as search. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 261–269, Online. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

Àlex Bravo, Luis Chiruzzo, Horacio Saggion, et al. 2018. Lastus/taln+ inco@ cl-scisumm 2018-using regression and convolutions for cross-document semantic linking and summarization of scholarly literature. In *BIRNDL@ SIGIR*.

Ziqiang Cao, Wenjie Li, and Dapeng Wu. 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)*, pages 132–138.

Ling Chai, Guizhen Fu, and Yuan Ni. 2020. Nlp-pingantech@ cl-scisumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 235–241.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224, Online. Association for Computational Linguistics.

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir Radev, Dayne Freitag, and Min-Yen Kan. 2019. Overview and results: Cl-scisumm shared task 2019.

Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1815–1826.

Arman Cohan, Luca Soldaini, and Nazli Goharian. 2015. Matching citation text and cited spans in biomedical literature: a search-oriented approach. In *proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 1042–1048.

Elnaz Davoodi, Kanika Madan, and Jia Gu. 2018. Clscisumm shared task: On the contribution of similarity measure and natural language processing features for citing problem. In *BIRNDL@ SIGIR*.

Dipanwita Debnath, Amika Achom, and Partha Pakray. 2018. Nlp-nitmz@ clscisumm-18. In *BIRNDL@ SIGIR*, pages 164–171.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radev. 2008. Blind men and elephants: What do citation summaries tell us about a research article? *Journal of the American Society for Information Science and Technology*, 59(1):51–62.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Rep4NLP@ACL*.

Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir Radev, and Min-Yen Kan. 2019. The cl-scisumm shared task 2018: Results and key insights.

Xiaoqi Jiao, Fang Wang, and Dan Feng. 2018. Convolutional neural network for universal sentence embeddings. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2470–2481, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hyonil Kim and Shiyan Ou. 2019. Nju@cl-scisumm-19. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 247–255. CEUR-WS.org.

Stefan Klampfl, Andi Rexha, and Roman Kern. 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)*, pages 122–131.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Lei Li, Junqi Chi, Moye Chen, Zuying Huang, Yingqi Zhu, and Xiangling Fu. 2018. Cist@ clscisumm-18: Methods for computational linguistics scientific citation linkage, facet classification and summarization. In *BIRNDL@ SIGIR*.

Lei Li, Yingqi Zhu, Yang Xie, Zuying Huang, Wei Liu, Xingyuan Li, and Yinan Liu. 2019. Cist@clscisumm-19: Automatic scientific paper summarization with citances and facets. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 196–207. CEUR-WS.org.

Tao Liu, Xin Wang, Chengguo Lv, Ranran Zhen, and Guohong Fu. 2020. Sentence matching with syntax- and semantics-aware bert. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3302–3312.

Shutian Ma, Heng Zhang, Jin Xu, and Chengzhi Zhang. 2018. Njust@ clscisumm-18. In *BIRNDL@ SIGIR*.

Shutian Ma, Heng Zhang, Tianxiang Xu, Jin Xu, Shaohu Hu, and Chengzhi Zhang. 2019. Ir&tm-njust @ clscisumm-19. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 181–195. CEUR-WS.org.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ofir Pele and Michael Werman. 2008. A linear time histogram metric for improved sift matching. In *European conference on computer vision*, pages 495–508. Springer.

Ofir Pele and Michael Werman. 2009. Fast and robust earth mover's distances. In *2009 IEEE 12th international conference on computer vision*, pages 460–467. IEEE.

Yoann Pitarch, Karen Pinel-Sauvagnat, Gilles Hubert, Guillaume Cabanac, and Ophélie Fraisier-Vannier. 2019. IRIT-IRIS at cl-scisumm 2019: Matching citances with their intended reference text spans from the scientific literature. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 208–213. CEUR-WS.org.

Animesh Prasad. 2017. Wing-nus at cl-scisumm 2017: Learning from syntactic and semantic similarity for citation contextualization. In *BIRNDL@ SIGIR (2)*.

Moreno La Quatra, Luca Cagliero, and Elena Baralis. 2019. Poli2sum@cl-scisumm-19: Identify, classify, and summarize cited text spans by means of ensembles of supervised models. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 233–246. CEUR-WS.org.

Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. The acl anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Bakhtiyar Syed, Vijayasaradhi Indurthi, Balaji Vasan Srinivasan, and Vasudeva Varma. 2019. Helium @ cl-scisumm-19 : Transfer learning for effective scientific research comprehension. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 214–223. CEUR-WS.org.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110, Sydney, Australia. Association for Computational Linguistics.

Narendra Babu Unnam, Krishna Reddy, Amit Pandey, and Naresh Manwani. 2022. Journey to the center of the words: Word weighting scheme based on the geometry of word embeddings. In *34th International Conference on Scientific and Statistical Database Management*, SSDBM 2022, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Pancheng Wang, Shasha Li, Ting Wang, Haifang Zhou, and Jintao Tang. 2018. Nudt@ clscisumm-18. In *BIRNDL@ SIGIR*.

Jen-Yuan Yeh, Tien-Yu Hsu, Cheng-Jung Tsai, and Pei-Cheng Cheng. 2017. Reference scope identification for citances by classification with text similarity measures. In *proceedings of the 6th international conference on software and computer applications*, pages 87–91.

Chrysoula Zerva, Minh-Quoc Nghiem, Nhung T. H. Nguyen, and Sophia Ananiadou. 2019. Nactem-uom @ cl-scisumm 2019. In *BIRNDL@SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 167–180. CEUR-WS.org.

Chrysoula Zerva, Minh-Quoc Nghiem, Nhung T. H. Nguyen, and Sophia Ananiadou. 2020. Cited text span identification for scientific summarisation using pre-trained encoders. *Scientometrics*, 125:3109–3137.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding.

A Background

A.1 SBERT

Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) , is a modification of the pre-trained BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) model. BERT is a popular attention mechanism-based model that takes a sentence (an arbitrary sequence of tokens)

as an input and learns contextual embeddings for each token in the sentence. Though BERT has achieved state-of-the-art performance in a wide variety of NLP tasks, its design renders it inappropriate for semantic similarity search and unsupervised tasks because BERT doesn't compute independent sentence embeddings and instead learns embeddings for each token of the sentence.

To overcome this problem, SBERT builds over the BERT's innovation of using a bidirectional encoder. SBERT leverages BERT-based siamese network architecture to embed sentences into a fixed-length vector by adding a pooling layer on top of the BERT layer. The SBERT siamese network architecture can be fined tuned using different losses such as triplet loss, contrastive loss, and cosine similarity loss. Moreover, SBERT is computationally inexpensive compared to BERT (Reimers and Gurevych, 2019).

A.2 Word Mover's Distance

Given pre-trained embeddings for the words, Word Mover's Distance (WMD) (Kusner et al., 2015) measures the distance between a pair of sentences (sequence of words). It exploits the underlying geometry of the word embeddings to represent a sentence as a weighted point cloud in the word embedding space. It formulates the problem of finding distance between two sentences as a transportation problem based on Earth Mover's Distance. It defines the dissimilarity between two sentences as the minimum amount of work (distance traveled) required to transport words from one sentence to the words of another sentence in the word embedding space. This minimum cumulative travel cost between words of two sentences is calculated by solving the following linear optimization problem.

$$\min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij} c(i,j)$$

$$\text{subject to:} \sum_{j=1}^{n} T_{ij} = s_i \quad \forall i \in \{1, ..., n\}$$

$$\sum_{i=1}^{n} T_{ij} = s'_j \quad \forall j \in \{1, ..., n\}$$

Here, s and s' are the normalized bag-of-words representation of two sentences. T is a flow matrix, where the $T_{ij} \geq 0$ entry indicates how much of word i in sentence s travels to word j in sentence s'. The total outgoing flow from a word i in sentence s to all the words j in sentence s' equals to the normalised frequency of word i, i.e ($\sum_{ij} T_{ij} = s_i$). The distance between two words in the embedding space is given by $c(i, j)$ and calculated using euclidean distance between the word embeddings. The final distance between two sentences is $\sum_{ij} T_{ij} c(i, j)$.

B Detailed Experimental Setup and Analysis

We demonstrate the performance of the proposed method on CL-SciSumm shared task (Jaidka et al., 2019; Chandrasekaran et al., 2019, 2020) task 1(a), where for each citance, we need to identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. These cited text spans are of the granularity of a sentence fragment, a full sentence, or several consecutive sentences (no more than 5). For this, we pick top k (we picked $k = 3$) semantically similar candidate cited sentences for a given citance by sorting their similarity scores. We evaluate the predictions against gold label annotations using F1 score.

We compare the performance of the proposed model CitRet, with the best 3 systems (of each category) submitted by NaCTeM-UoM (Zerva et al., 2019) and the best 2 systems submitted by team NLP-PINGAN-TECH (Chai et al., 2020), over CL-SciSumm test set.

The systems submitted by NaCTeM-UoM are based on BERT. Along with *BERT 2018/19 OV + 2018 FT* (a BERT model fine-tuned on the CL-SciSumm 2018-2019 dataset), they submitted models *ACL 2018* and *SciBERT 2018*. Both these models are first trained on significantly large domain-specific corpora and then fine-tuned on CL-SciSumm dataset. *ACL 2018* is trained ACL-ARC (Radev et al., 2013) whereas *SciBERT 2018* is based on SciBERT model (Beltagy et al., 2019), which is pre-trained on collection of 1.14M documents from Semantic Scholar (Ammar et al., 2018).

NLP-PINGAN-TECH team also centered their approach around fine-tuning BERT-based models using larger domain-specific datasets. Their best performing system *SciBERT-SemBERT* is an ensemble of SciBERT, SemBERT (Zhang et al., 2020) based on SciBERT, *Sci-BERT-fake-token* (tokens for position and section details like *[method][sid=xx][ssid=xx]* are added as prefixes to the sentences) and *SciBERT-special-token* (tokens for position and section details like

[method],[sid=1], etc. are added to the SciBERT dictionary to avoid split during tokenization). The other method *SciBer-ACLBERT*, submitted by the NLP-PINGAN-TECH team that achieved a high score, also leverages SciBERT and ACL corpora.

In comparison, the proposed model is trained only on the CL-SciSumm training dataset that consists of 40 manually annotated articles, which were used in the 2018 CL-SciSumm challenge as well, and 1000 document sets that were automatically annotated using neural networks. These 1000 document sets were introduced in 2019 and are of lower quality compared to the manually annotated dataset. Also, we do not use any external corpora to fine-tune our model.

Table 1 shows the performance comparison of our model with the SOTA models. The last 4 rows show the ablation study of our pipeline marked with †. It can be observed that fine-tuning SBERT using our strategy resulted in better scores than *BERT 2018/19 OV + 2018 FT*, *ACL 2018* and *SciBERT 2018*. All three models are based on BERT and SciBERT and trained on much larger datasets. This shows that learning sentence embeddings instead of token embeddings performs better for textual similarity tasks (Reimers and Gurevych, 2019).

Moreover, as evident from the ablation study, individual components of our pipeline also help in increased performance. The most significant improvement, of 30% over fine-tuned SBERT, was achieved by weighted contextual embeddings (*SBERT + WCE*). It can be noted from Table 1 that using just *SBERT + WCE* component of our pipeline outperformed all the SOTA models. This empirically validates that utilizing the unique structural characteristics of the scientific documents can significantly improve the results. Further denoising the weighted contextual embeddings (SBERT + WCE + D) for $m = 3$ improved the performance by around 5%. Moreover, augmenting the contextual embeddings-based similarity scores with WMD achieved a new SOTA by advancing the results of *SBERT + WCE + D* by over 9%.

We also performed experiments to check how the performance of the model varies with train and test sets' size. The proposed method showed improvement when we used 1000 document sets that were automatically annotated using neural networks along with the 40 manually annotated documents. We obtained 0.17790 F1 (0.1869 Recall and 0.1697 Precision) when we trained with just the manually annotated dataset that contained only 40 documents. We also experimented with the 1000 noisy training samples by randomly splitting them into the train (80%) and test (20%) sets and obtained 0.2779 F1 (0.4836 Recall and 0.195 Precision).

C Implementation details

We preprocess the sentences by lowercasing the words, removing the stopwords, removing the special characters and errors due to OCR using NLTK library and regex functions. For SBERT we use the implementation provided by Reimers and Gurevych (2019). We train the model for 3 epcohs with a batch size of 16 on Nvidia GeForce GTX 1080 Ti GPU. The total training time is around 7 minutes. We use AdamW optimizer with a learning rate of $2e^{-05}$ and weight decay of = 0.01. For WMD we use Gensim's implementation (Kusner et al., 2015; Pele and Werman, 2008, 2009). The code is available at `https://github.com/AmitPandey-Research/CitRet_Public.git`.