UniHD@CL-SciSumm 2020: Citation Extraction as Search

Dennis Aumiller[*], **Satya Almasian**[*], **Philip Hausner**[*], **Michael Gertz**
Heidelberg University, Heidelberg, Germany
{lastname}@informatik.uni-heidelberg.de

Abstract

This work presents the entry by the team from Heidelberg University in the CL-SciSumm 2020 shared task at the Scholarly Document Processing workshop at EMNLP 2020. As in its previous iterations, the task is to highlight relevant parts in a reference paper, depending on a citance text excerpt from a citing paper.

We participated in tasks 1A (cited text span identification) and 1B (citation context classification). Contrary to most previous works, we frame Task 1A as a search relevance problem, and introduce a 2-step re-ranking approach, which consists of a preselection based on BM25 in addition to positional document features, and a top-k re-ranking with BERT. For Task 1B, we follow previous submissions in applying methods that deal well with low resources and imbalanced classes.

1 Introduction

Scientific papers are among the most important means of communication between researchers that enable scholars to share their knowledge and progress, and provide other scientists with retrospective documentation and starting points for further improvements. A crucial part of this scientific exchange is citations, which refer to prior academic work that helped researchers put their work in the context of a broader scientific vision. The CL-SciSumm Shared Task aims to construct meaningful summarization of this scientific communication by utilizing information extracted from such citations. The key contributions of a paper are identified by investigating which parts of the paper are cited, since citations usually highlight the critical points and main contributions of a paper. Additionally, citations often target several aspects of a paper, and hence, can complement each other (Jaidka et al., 2018a). As a result, a paper's contributions may be outlined by summarizing the parts of the paper that other researchers cited.

Another essential aspect of citations is the manner in which they cite another work: Some may refer to results obtained in previous work, some build on top of the reference paper's methodology or propose modifications, and some debate claims hypothesized in a prior paper (Teufel et al., 2006). Therefore, particular citations of the same paper may refer to different text spans in various sections of the reference paper. If the authors use the cited work as a basis or starting point, they often refer to the methodology section. At the same time, a citation comparing the goals or results with that of prior work mainly refers to the introduction or evaluation of a paper.

Building on the ideas presented above, the CL-SciSumm Shared Tasks (Jaidka et al., 2016, 2017, 2018b; Chandrasekaran et al., 2019, forthcoming) split up the task of scientific summarization into multiple sub-tasks. These sub-tasks are formulated as follows: Given a set of reference papers (RP) and a set of corresponding citing papers (CP) that contain citations to one of the reference papers, and in which text spans (so called citances) have been identified that pertain to a particular citation to the respective RP, participants of the Shared Task have to develop methods to solve the following tasks:

- *Task 1A:* For each text span around a citation (citance), a span in the RP has to be identified that most accurately reflects the citance. Spans may be a sentence fragment, a complete sentence, or up to 5 consecutive sentences.
- *Task 1B:* Each citance has to be classified based on its citation context. The five facet categories are *Aim, Hypothesis, Implication, Method*, and *Results*. Additionally, a cited text span may belong to more than one facet.

[*] These authors contributed equally to this work.

Proceedings of the First Workshop on Scholarly Document Processing, pages 261–269
Online, November 19, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

- *Task* 2*:* The final task is to generate a summary of the RP, based on the cited text spans, with a word limit of 250. Task 2 is optional.

Our team participated in Tasks 1A and 1B, and hence, we do not construct a final summarization of the respective RPs. As the quality of a pre-selection can significantly improve the results of downstream tasks (Liu and Lapata, 2019), we focus primarily on improving selection results in Tasks 1A and 1B. We formulate Task 1A as a search problem modeled in two steps: First, a set of cited sentences is extracted by employing a search using BM25 in combination with the sentence position in the document. Here, the query term is the citation itself, and each sentence in the reference paper is treated as a single document in the search process. In the second step, a top-k re-ranking is applied that utilizes BERT to extract the most relevant sentences. For Task 1B, we follow previous work by Zerva et al. (Zerva et al., 2019) in implementing one-versus-rest classifiers, but base them on perceptron classifiers instead of random forests or BERT-based models.

2 Related Work

In previous editions of the CL-SciSumm Shared Task, various effective strategies were proposed to solve Task 1 (Chandrasekaran et al., 2019). To find the relevant reference sentence, most systems from 2019 focused on sentence similarity. Similarities are either obtained by methods such as TF-IDF and Jaccard or embedding-based methods to mine more semantic information (Pitarch et al., 2019) or by designing specific features and learning sentence similarities in a supervised manner (Li et al., 2019; Chiruzzo et al., 2019). Task 1A can also be framed as a classification task and solved via a single or ensemble of multiple classifiers. An ensemble of regression and classification models are trained on the reference and citation sentences (Quatra et al., 2019; Ma et al., 2019). The best performing system from 2019 (Zerva et al., 2019) uses a BERT-based model to solve the task in two ways, first by using sentence similarity of BERT vectors trained on citation and reference sentence pairs, and second by using bilateral multi-perspective matching model. The authors also perform extensive data cleaning and mine additional data from the PDF version of the papers to fine-tune their language model. The most similar work to our approach is by (Kim and Ou, 2019), where the authors propose a two-stage similarity-based unsupervised ranking method. In the first stage, they use modified Jaccard similarity to select the top-5 relevant sentences. These top-5 selected sentences are ranked again using a listwise ranking model and the features from the first stage. In contrast, we utilize a variant of BM25 for our original ranking, a larger candidate set of 10 sentences, and a pair-wise ranking for our second stage using BERT. Since neural models pretrained on language modeling such as BERT have achieved impressive results for several NLP tasks, many have applied them to search-related tasks. BERT is used in ad-hoc document ranking (MacAvaney et al., 2019; Yang et al., 2019) and also for multi-stage ranking, where the original ranking is often performed by an efficient method like BM25 and the results are ordered by their relevancy score by single or multiple re-ranking stages (Nogueira and Cho, 2019; Nogueira et al., 2019).

3 Methodology

In our approach, we aim to solve Task 1A and 1B independently. We formulate the citation linkage in Task 1A as a search problem, where the CP sentence is the query, and each sentence in the respective RP is considered a separate “document” in an indexed collection. For Task 1B, we mainly follow existing work to deal with the unbalanced data and a low number of samples. In the following, we explain the framework in more detail.

3.1 Task 1A

The citation linkage module consists of three main stages: 1) Retrieval of the top-k relevant sentence to a given citation from the reference paper by a standard search mechanism, such as BM25. 2) Re-ranking the retrieved sentences using a more computationally expensive model, such as BERT re-ranker (Nogueira and Cho, 2019), and 3) choosing the relevant candidates as answers based on thresholding of re-ranking scores. The full pipeline is shown in Figure 1. In the ranking stage, the reference paper is indexed using Apache Solr, and candidate sentences are generated by querying with the citation sentence. The BERT re-ranker filters the set to only relevant sentences, and the facet classifier predicts the respective facets for them.

3.1.1 Ranking

While submissions to previous iterations of the workshop already considered a wide range of similarity metrics, such as TF-IDF, Word2Vec (Pitarch

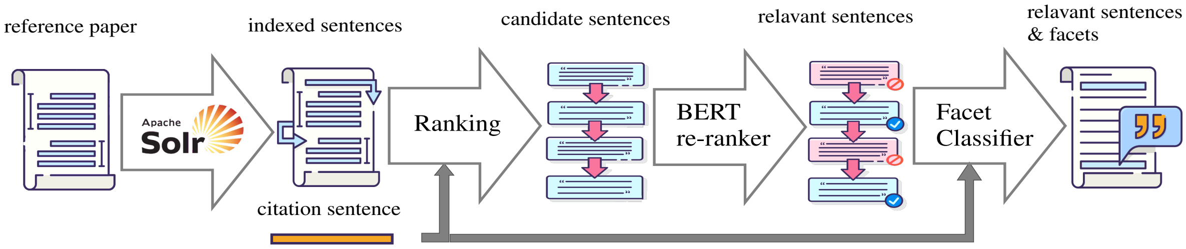


Figure 1: Overview of the proposed pipeline. Documents are indexed in Apache Solr, and a candidate ranking set is formed by querying with the citing span. A BERT-based re-ranking module chooses the relevant sentences based on the candidate set, on which facet prediction is performed.

et al., 2019), or learned similarities (Zerva et al., 2019; Syed et al., 2019), there is only the approach by (Kim and Ou, 2019) that similarly phrases the problem as a 2-step ranking problem. Specifically, we similarly treat each reference paper as a "document collection" of its pre-segmented sentences, and construct a search index using Apache Solr[1]. Per default, Solr implements the Lucene variant of BM25 detailed in (Kamphuis et al., 2020) as its scoring function. BM25 is in our opinion more suitable than related measures, such as TF-IDF, as it considers both sentence lengths, as well as the full input query, instead of single terms. We chose the entire 40 annotated documents from the 2018 training data to tune parameters. Feature selection as well as the indexing step itself is unsupervised, and thus requires no further splitting into training and validation set. Experiments with alternative weighting metrics (we compared TF-IDF, DFR (Amati and van Rijsbergen, 2002), and IBS (Clinchant and Gaussier, 2010)) yielded worse results and were discarded.

Preprocessing Aside from the scoring function, Solr's indexing modules allow for a custom preprocessing pipeline of both indexed documents and submitted queries. We performed experiments varying the following functions over different search fields: 1) Stopword filtering, 2) keyphrase filtering, 3) lowercasing, 4) porter/Snowball stemmer, 5) synonym filter, 6) word delimiter filter, and 7) shingling. For stopwords, we used the English stopwords provided by Solr. Synonyms were manually generated by comparing citances and references of the 2018 training corpus and consist mostly of spelling variations (e.g., "co-occurrence" and "cooccurrence") or community-specific abbreviations (e.g., "HMM", "Hidden Markov Model").

Additional Document Features While Li et al. utilized further features such as (relative) section position (Li et al., 2019), we found that the pre-segmented data contained insufficiently accurate section annotations. Instead, we chose to only incorporate the relative sentence position, defined as $\frac{\text{sentence id}}{\#\text{sentences}}$, since it is in theory more tolerant to incorrect segmentations. Specifically, we boost the ranking scores of query results that appear in the first 30% of the document by 1.5 times of their original score. Further features did not improve results in our experiments.

Text/Query Formatting As the quality of results depends on the quality of the input, rule-based preprocessing was employed to clean both the indexed content and query strings. Mainly, indicators of citations from the citances were deleted, as they do not relate to the sentence content, and results degrade by leaving them in. Furthermore, "math-like" text, such as formulas or vector representations, for both indexed sentences and queries were masked with special tokens <COMPLEXITY>, <PROBABILITY>, <FUNCTION> or <VECTOR>. The masked tokens are protected from tokenization by Solr.

Ensembling To emphasize the generalization of our ranking module, we finally ensemble several text fields with varying configurations. Aggregation of results is performed by first retrieving the top-k results including their ranking scores for each of the text fields in the ensemble. We merge individual query results by summing up ranking scores, and return the re-ordered top-k results of the aggregated candidate set. These are then handed over to the re-ranking module. Note that in this scenario, a sentence can be deemed relevant if it only appears in a single query result, but with a sufficiently high score. In our experiments, we chose $k = 10$.

[1] https://lucene.apache.org/solr/; we used version 8.5.2 in our experiments.

Baseline Approach Since the BERT re-ranker performs the final restriction of candidates, we also wanted to compare to a direct restriction of candidates through Solr only. For this, we return only four results per model in an ensemble. We then employ simple majority voting over the individual model results to return a final set of candidates.

3.1.2 Re-ranking with BERT

The re-ranker module estimates a relevance score of sentence s_i for each pair of candidate passage and query. In our case, the candidate passage is an arbitrary sentence in the reference paper, denoted as rs_i, and the query is the citing sentence, cs_i. The candidate sentences from the reference paper are generated by the ranker described in the previous section. The re-ranker than takes the output of the ranking module and learns which of the top-k results are relevant to the citation sentence. To compute the relevance score we use the BERT re-ranker (Nogueira and Cho, 2019), which uses the pre-trained deep bidirectional language model, BERT (Devlin et al., 2019), to learn relevance patterns. There exist multiple variations of the BERT architecture for re-ranking; However, we chose the simple addition of one linear layer on top of the BERT representation. The simple one layer re-ranker is proven to be most effective in comparison to more complex architectures, where instead of the last layer representation combination of different intermediate layers are used for re-ranking (Qiao et al., 2019). Following the same notation as (Devlin et al., 2019), we feed in the citation sentence, cs_i, as sentence A and reference sentence, rs_i, as sentence B. We truncate the sentence from the reference paper, so that the concatenation of cs_i and rs_i results in at most 512 tokens. We add a classification layer on top of BERT$_{BASE}$ for binary classification and use the [CLS] vector as the input to the classification layer. The [CLS] vector encodes the joint representation of the citation sentence and the reference sentence, and the classification layer computes the relevance probability for each reference sentence independently. The final list of the relevant sentences is obtained by sorting the candidates based on the relevance probability. We fine-tune the pre-trained BERT model using cross-entropy loss as follows:

$$L = \sum_{j \in J_{pos}} log(s_j) - \sum_{j \in J_{neg}} log(1 - s_j) \quad (1)$$

where J_{pos} contains the set of all relevant reference sentences, and J_{neg} are the negative examples, retrieved from the top-10 sentences by BM25. The final set of relevant sentences is computed by thresholding the relevance probability.

We acknowledge the possibility of mixing results with the pre-selection scores returned by BM25, but argue that a recall-optimized tuning of BM25 would likely not improve hard instances that are generally based on semantic similarity, rather than syntactic similarities. Furthermore, we also did not experiment with pair-wise losses, as the query lengths exceed those of traditional IR setups (Nogueira et al., 2019), and subsequently triplets required for training are frequently longer than the 512 token limit of BERT.

3.2 Task 1B

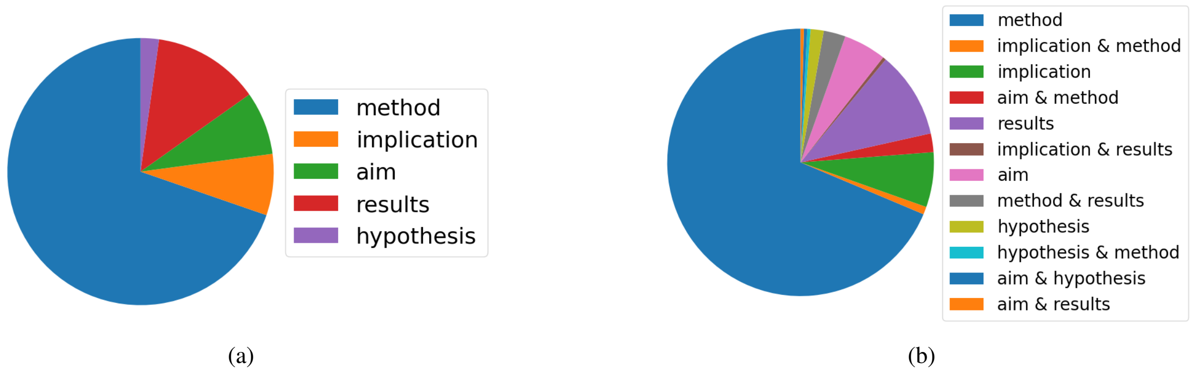
Task 1B aims to extract discourse facets from the given citation spans. We formulated this as a multi-class and multi-label classification task, in which each of the five predefined facets (*Aim, Hypothesis, Implication, Method*, and *Result*) is one class. A first investigation of the given data reveals two challenges relevant to this task. First, the data set consists of only 753 samples, which is a small number of instances to train a machine learning model. Second, there is a significant imbalance in the distribution of labels, as seen in Figure 2. To overcome these challenges, we framed the problem as multiple binary classification tasks by employing five one-vs-rest perceptron classifiers. We then extracted features for the classification as follows:

- First, all words in the citation sentences and the reference sentences predicted in Task 1A are lemmatized, and stop-words are removed using NLTK's stop word list.

- Word-level uni- and bigrams are extracted from all reference and citation sentences. Following this, a bag-of-words model is constructed, and TF-IDF scores are calculated.

While these were the final (and only) features of the constructed classifier, we experimented with the following features as well, which unfortunately did not improve results:

- Since the position of a sentence in the document should be meaningful, we integrated the sentence ID as a feature.



(a)

(b)

Figure 2: Distribution of discourse facets among the annotated data set when (a) counting all facets independently, and (b) counting combinations of discourse facets as well.

- As a follow-up, multiple common sections within a scientific paper were defined: *Abstract, Introduction, Related Work, Methods, Results, Conclusion, Acknowledgments* while remaining sections were labeled as *Unknown*. We constructed a mapping from the 50 most occurring section titles within the training data set to one of the 7 section types. The section ID then replaced the sentence ID that was used in the previous approach as a feature.

- In another attempt, the imbalance of the data set was targeted by sampling at most 100 samples from each discourse facet class. This experiment resulted in a situation in which the rare discourse facets classification accuracy improved. However, precision and recall for the most common facets (*Method* and *Results* citations) dropped significantly. Hence, this approach was not pursued any further.

4 Experimental Settings and Evaluation

Aside from the pure evaluation of results, we further verified the correctness of the existing training data. For the manually annotated samples from 2018, we were able to identify several citations that were either duplicated or mismatched (differing reference text and reference offset, or similar for the citance). Further, we deleted around 30 empty documents from the Scisummnet corpus (containing non-empty "sentences" for less than 10% of the document). A pull request with several changes is currently awaiting approval for the main shared task repository.

Feature	Field 1	Field 2	Field 3
Tokenizer	Standard	Whitespace	Standard
Lowercase	Yes	Yes	Yes
Word Filters	Possessive	Delimiter	No
Stemming	No	Porter	Snowball
Shingling	Bigram	Bigram	Trigram
Stopwords	Yes	Yes	No
Synonyms	Yes	Yes	Yes

Table 1: Features for the text fields used in the ensembles. Configurations differ significantly to ensure a heterogeneity of results for better generalization.

4.1 Task 1A

As detailed in Section 3.1.1, we experimented with several combinations for our ensemble models. Specifically, we ended up with three different text fields, which we combined into two ensembles. The first ensemble consists of fields 1&2 (*2-field*), and another of all three fields (*3-field*) Specific configurations of the fields are detailed in Table 1. As mentioned previously, the performance is judged by evaluation of the manually annotated documents from the 2018 corpus.

4.1.1 Re-ranking with BERT

To train the BERT re-ranker, we use the combination of manually and automatically annotated data. To make our result comparable to Zerva et al. (Zerva et al., 2019), we held out the same eight articles from the manually annotated articles as a preliminary test set[2]. The training data is provided by the Solr ranker in two formats, resulting in two training strategies. In the first variation, the

[2] 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

top-10 sentences are ranked with Solr, using the 2-field or 3-field ensembles, respectively, and relevant sentences among the retrieved data are marked a positive example. The remaining examples are considered negative examples. One drawback of this approach is that if Solr mistakenly misses one of the relevant sentences, it does not appear in the candidate set for re-ranking in the first place. Therefore, the re-ranker cannot incorporate the missing positive examples in the training process. To overcome this shortcoming, we propose the supervised candidate set, in which we manually added any missing true positive example from the ground truth data to the candidate set. The second approach is denoted by subscript S in the evaluation runs. All models are trained for 3 epochs with a batch size of 52 on two TITAN RTX GPUs, with $24GB$ of RAM. Training takes approximately 3.5 hours. We used Adam as the optimizer for our network with a learning rate of $5\text{e}{-}7$. Moreover, to avoid exploding gradients, we clipped gradients larger than 0.5. The final list of relevant sentences is generated by re-ordering the sentences based on the relevance score and choosing the top-4 sentences. Experiments to return results based on fixed thresholds yielded much lower precision, and indicate that the model may not yet properly normalize scores across samples.

Model	Precision	Recall	F1
Solr$_2$	0.128	0.217	0.161
Solr$_3$	0.124	0.217	0.158
(Zerva et al., 2019)	0.171	0.334	0.226
BERT$_{S2}$	0.084	0.239	0.124
BERT$_{S3}$	0.077	0.219	0.114
BERT$_2$	0.087	0.248	0.129
BERT$_3$	0.095	0.270	0.141

Table 2: Precision, recall, and F1-score on the preliminary test set used by (Zerva et al., 2019), for which we also report their best-performing model. Subscript numbers describe the number of ensemble fields for ranking. Subscript S indicates the supervised variant, where the ground truth is always added to the preselection candidate set.

Discourse Facet	Precision	Recall	F1	#samples
Aim	1.000	0.333	0.500	18
Hypothesis	0.000	0.000	0.000	1
Implication	0.000	0.000	0.000	21
Method	0.742	0.969	0.841	98
Results	0.444	0.800	0.571	15
Micro Average	0.685	0.739	0.711	153
Macro Average	0.437	0.421	0.382	153

Table 3: Precision, recall, F1-score, and number of samples in the validation set for each discourse facet as well as micro and macro average.

4.1.2 Analysis of Results

Table 2 shows the precision, recall, and F1-score on the held-out preliminary test set, and re-ranking based on BERT degrades the performance. Since the recall could not be improved by re-ranking, our main objective was to obtain better precision. However, BERT fails to learn to improve upon that. We attribute this failure to the limited training data and noisy annotation of the automatically annotated data, as well as a small set of positive vs. negative sentences on which the classifier was trained. Despite the additional re-ranking step, our best performing models are the original Solr ranking top-4. Surprisingly, the models trained on supervised candidate sets have significantly lower precision, indicating that the additional information from the missing examples can be misleading to the final re-ranking. One reason for the weak performance could be the difference in the distribution of the training and test set imposed by adding the additional ground truth values, as these results do not initially show up in our first step ranking procedure. Furthermore, the majority of the seen samples during training time consists of the automatically extracted citation spans by Scisummnet (Yasunaga et al., 2019), which significantly differs from extracted portions on the manually annotated data. Moreover, high-quality results form the ranking step can also influence the effectiveness of our two-stage retrieval. The 3-field ensemble produces a better original ranking and candidate set, resulting in slightly better relevancy scores.

4.2 Task 1B

For evaluation of task 1B, the data set of the 40 manually annotated papers are randomly split into a training set consisting of about 80% of the data, and a validation set consisting of the remaining 20% of annotations. Table 3 shows the results of the trained model. As shown, recall of discourse facets is reasonable for classes *Method* and *Results*. For facets less represented in the data set, however, the model performs poorly. This imbalance in the data set is also observed in the micro and macro average: The micro average indicates a much better performance than the macro average, since the majority of discourse facets in the data set are of type

Model	SO F1	ROUGE F1	1B F1
Solr_2	**0.161**	0.113	0.292
Solr_3	0.153	0.107	0.294
(Zerva et al., 2019)	0.126	0.075	0.312
(Wang et al., 2018)	0.145	**0.131**	0.262
(Li et al., 2019)	0.106	0.034	**0.389**
(Li et al., 2018)	0.122	0.049	0.381
BERT_{S2}	0.122	0.059	-
BERT_{S3}	0.110	0.055	-
BERT_2	0.122	0.059	-
BERT_3	0.118	0.059	-

Table 4: Task 1A Sentence Overlap F1 (SO F1), task 1A ROUGE F1, and task 1B F1 on the official test set. For comparison, we report the best-performing models for Task 1A (based on SO F1) and task 1B from the 2018 and 2019 shared tasks, which were evaluated on the same test set.

method, which is classified correctly more often than the rare occurrences of other classes.

4.3 Official Test Results

Table 4 shows the results of our models on the 2018 test set, used as the official benchmark for this year's iteration as well. Comparing our results to the best-performing models from the past two years, we can see a clear improvement of Sentence Overlap F1. Surprisingly, our model shows a much better generalization to the official test set than Zerva et al., based on their results shown in Table 2. However, despite a better Sentence Overlap F1, our ROUGE is still trailing behind the submission by Wang et al., which leaves open questions regarding the best optimization criterion. For task 1B, our results are significantly lacking behind previous best-performing entries, however, we would be interested how our improved selection of sections would affect their respective predictions.

5 Future Work

As indicated in the results on the different test sets for Task 1A, a high-quality sub-selection of potentially relevant sections can significantly boost the performance of more learning-based methods. Despite some optimization, we still saw relatively low recall values for some citation spans, which can have several causes. We believe that improvements to the initial recall can still significantly boost results, but require measures that do not entirely rely on simple word frequency measures such as BM25 or TF-IDF. While the re-ranker seemed to struggle with the limitations of the current setup, future exploration with different re-ranking approaches might ultimately yield improvements over the results returned by Solr. We intend to utilize contextualized results, or "snippets" of several sentences within a single result in future submissions to the workshop to increase pre-selection recall. Similarly, for Task 1B, replacing prediction for low-resource classes with rule-based approaches could balance classification scores on the unseen test set.

References

Gianni Amati and C. J. van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389.

M. K. Chandrasekaran, G. Feigenblat, Hovy. E., A. Ravichander, M. Shmueli-Scheuer, and A De Waard. forthcoming. Overview and insights from scientific document summarization shared tasks 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*.

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir R. Radev, Dayne Freitag, and Min-Yen Kan. 2019. Overview and Results: CL-SciSumm Shared Task 2019. In *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 153–166. CEUR-WS.org.

Luis Chiruzzo, Ahmed AbuRa'ed, Àlex Bravo, and Horacio Saggion. 2019. LaSTUS-TALN+INCO @ CL-SciSumm 2019. In *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 224–232. CEUR-WS.org.

Stéphane Clinchant and Éric Gaussier. 2010. Information-based models for ad hoc IR. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 234–241. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT , Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Devanshu Jain, and Min-Yen Kan. 2017. The CL-SciSumm Shared Task 2017: Results and Key Insights. In *Proceedings of the Computational Linguistics Scientific Summarization Shared Task (CL-SciSumm 2017) organized as a part of the 2nd Joint*

Workshop (BIRNDL) and co-located with the 40th International ACM, volume 2002 of *CEUR Workshop Proceedings*, pages 1–15. CEUR-WS.org.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2016. Overview of the CL-SciSumm 2016 Shared Task. In *Proceedings of the Joint Workshop (BIRNDL) co-located with the Joint Conference on Digital Libraries 2016 (JCDL)*, volume 1610 of *CEUR Workshop Proceedings*, pages 93–102. CEUR-WS.org.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2018a. Insights from CL-SciSumm 2016: the faceted scientific document summarization Shared Task. *Int. J. on Digital Libraries*, 19(2-3):163–171.

Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir R. Radev, and Min-Yen Kan. 2018b. The CL-SciSumm Shared Task 2018: Results and Key Insights. In *Proceedings of the 3rd Joint Workshop (BIRNDL) co-located with the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, volume 2132 of *CEUR Workshop Proceedings*, pages 74–83. CEUR-WS.org.

Chris Kamphuis, Arjen P. de Vries, Leonid Boytsov, and Jimmy Lin. 2020. Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 28–34. Springer.

Hyonil Kim and Shiyan Ou. 2019. NJU@CL-SciSumm-19. In *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 247–255. CEUR-WS.org.

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 *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018) co-located with the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018), Ann Arbor, USA, July 12, 2018*, volume 2132 of *CEUR Workshop Proceedings*, pages 84–95. CEUR-WS.org.

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 *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 196–207. CEUR-WS.org.

Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL, Volume 1: Long Papers*, pages 5070–5081. Association for Computational Linguistics.

Shutian Ma, Heng Zhang, Tianxiang Xu, Jin Xu, Shaohu Hu, and Chengzhi Zhang. 2019. IR&TM-NJUST @ CLSciSumm-19. In *Proceedings of the 4th Joint Workshop (BIRNDL)= co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 181–195. CEUR-WS.org.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42nd International ACM SIGIR*, pages 1101–1104. ACM.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR*, abs/1901.04085.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *CoRR*, abs/1910.14424.

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 *Proceedings of the 4th Joint Workshop (BIRNDL) 2019 co-located with the 42nd International ACM SIGIR)*, volume 2414 of *CEUR Workshop Proceedings*, pages 208–213. CEUR-WS.org.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *CoRR*, abs/1904.07531.

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 *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 233–246. CEUR-WS.org.

Bakhtiyar Syed, Vijayasaradhi Indurthi, Balaji Vasan Srinivasan, and Vasudeva Varma. 2019. Helium @ CL-SciSumm-19 : Transfer Learning for Effective Scientific Research Comprehension. In *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM 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 *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110. ACL.

Pancheng Wang, Shasha Li, Ting Wang, Haifang Zhou, and Jintao Tang. 2018. NUDT @ clscisumm-18. In *Proceedings of the 3rd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2018) co-located with the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018), Ann Arbor, USA, July 12, 2018*, volume 2132 of *CEUR Workshop Proceedings*, pages 102–113. CEUR-WS.org.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple applications of BERT for ad hoc document retrieval. *CoRR*, abs/1903.10972.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI*, pages 7386–7393. AAAI Press.

Chrysoula Zerva, Minh-Quoc Nghiem, Nhung T. H. Nguyen, and Sophia Ananiadou. 2019. NaCTeM-UoM @ CL-SciSumm 2019. In *Proceedings of the 4th Joint Workshop (BIRNDL) co-located with the 42nd International ACM SIGIR*, volume 2414 of *CEUR Workshop Proceedings*, pages 167–180. CEUR-WS.org.