A Global Context-aware Online Citation Recommender: Deep Neural Modules and the Role of Structural Information

Carolin Schindler*
carolin.schindler@uni-ulm.de
Ulm University, Germany

Danial Podjavorsek*
danial.podjavorsek@uni-ulm.de
Ulm University, Germany

Simon Birkholz*
simon.birkholz@uni-ulm.de
Ulm University, Germany

Marcel Hoffmann
marcel.hoffman@uni-ulm.de
Ulm University, Germany

Nicolas Lell
nicolas.lell@uni-ulm.de
Ulm University, Germany

Ansgar Scherp
ansgar.scherp@uni-ulm.de
Ulm University, Germany

ABSTRACT

Citing utilized work correctly, appropriately, and exhaustively is a challenging task that is faced by students and researchers writing a scientific paper. To assist researchers in terms of when and what to cite while writing a paper, we propose the modular end-to-end global context-aware citation recommendation pipeline SCOPE. Thereby, we employ state-of-the-art deep neural network components performing cite-worthiness detection and citation recommendation. The citation recommendation task is performed in two subsequent steps: A global citation recommender prefetches candidate papers that are finally ranked by a local citation recommender. Since citing behaviour is section-dependent, we additionally investigate the effect of incorporating these information into our pipeline. To this end, we perform a structure analysis through which we obtain templates for section sequences and a synonym dictionary of section headings, allowing us to map section headings to a predefined set of section types. We conducted experiments on the basis of the S2ORC dataset and LaTeX documents crawled from arXiv. Applying the results of our structure analysis, 56% of the papers in the arXiv dataset can be mapped to unique templates by inferring the *Method* section. Explicitly adding the section information to the neural networks has little effect. However, utilizing more local information on the section- or paragraph-level improves the performance of the pipeline components and baselines. Our composition of SCOPE with neural networks leaves room for improvement, achieving an accuracy of $A@5 = 56.86\%$ on the arXiv dataset. We make our implementation of the pipeline SCOPE publicly available: https://github.com/Data-Science-2Like/SCOPE

KEYWORDS

Citation Recommendation, Structure Analysis, Cite-worthiness Detection, Automated Writing Evaluation

1 INTRODUCTION

An essential part of scientific writing is to cite utilized work correctly, appropriately, and exhaustively. This not only avoids plagiarism, but gives credit to the respective researchers and makes the writing more persuasive [43]. However, citing is a challenging task [35] including, for instance, when to cite and what to cite [56]. With when to cite, we refer to the task of cite-worthiness detection [75], i. e., whether a portion of text requires a citation or not. The question what to cite is tackled by citation recommendation [22], which aims to suggest sources in order to substantiate pieces of written text. This is different from the task of paper recommendation [6, 41], where the user searches – based on already reviewed papers – for further literature that is interesting to their topic and hence worth reading.

We focus on supporting students and researchers in the domain of computer science (especially in the fields of artificial intelligence and machine learning) in terms of when and what to cite by employing a citation recommendation pipeline. In this domain, scientific works are commonly written in LaTeX, allowing to extract information easily since the structured source files are available. The scientific work we are recommending citations for is the *citing paper*, whereas the paper getting cited is the *cited paper*. The position where a citation is placed in the citing paper is denoted by a *citation marker*. The portion of surrounding text, which is intended to be supported by the citation, is the *citation context* [61]. There are two major kinds of citation recommendation that can be distinguished: global versus context-aware (also referred to as local) citation recommendation. The former aims at retrieving citations as references without backing up a specific citation context. Thereby, it utilizes the title, abstract, or even the complete text of the citing document including the so-far existing reference list. The latter only makes use of the citation context as a query, therefore, it is already known where the recommended citation should be placed within the document.

Most existing context-aware citation recommenders [22] only deal with what to cite by assuming that the scientific work is already written in its entirety, and hence the existing citations are used as a ground-truth for when to cite. However, this is not an appropriate scenario where students are currently in the process of writing a paper and might require support in terms of when to cite as well. Furthermore, as the work is still in progress, commonly utilized global information, such as the abstract and the title, are not guaranteed to be available or in their final state for the recommendation. In addition, it is known that the citing behaviour is section-dependent [10]. More precisely, this means that the amount of citations and the recency of cited papers is correlated with the sections in the citing paper.

Yet, these findings have not been incorporated into a citation recommender. Hence, it is unclear to which extend a section-specific

*The authors contributed equally to this research.

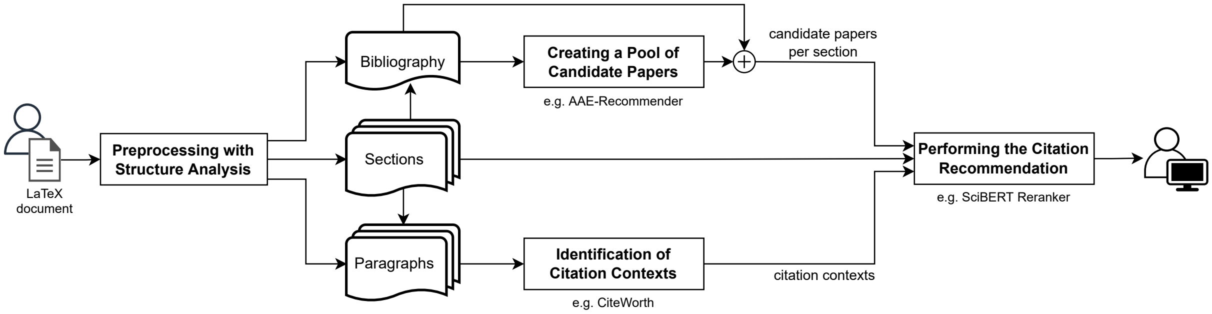

Figure 1: Structure of our proposed online pipeline SCOPE for global context-aware citation recommendation.

recommendation improves the performance. Since the section headings and structure vary across different papers, one needs to define a mapping to a common set of section types in order to be able to utilize the aforementioned correlations. There have already been successful efforts in including the paradigm of cite-worthiness detection into the citation recommendation pipeline [23, 29, 33] and in building systems for online citation recommendation [23, 33, 44]. The latter consider the fact that the scientific writing is still ongoing and the recommendation needs to be performed fast. Nevertheless, these systems do not employ deep neural network components in their pipeline, which are currently state-of-the-art for individual sub-tasks while posing a risk to the real-time functionality of the system.

In this work, we implement a modular end-to-end global context-aware citation recommendation pipeline which is aimed for an online use during the process of writing. This pipeline consists of state-of-the-art deep neural network components and a module for structural analysis of the scientific document, allowing to enrich the representation of the citing document with global section information. It is important for us to identify citation contexts in the citing document that are actually cite-worthy [24, 29, 75], instead of suggesting citations for all possible contexts. In order to recommend a cited paper, we need to rank a list of *candidate papers*, which form the pool of potentially cited papers. To this end, we have to efficiently retrieve the candidate papers as a subset of all possible papers by means of a prefetching module prior to the actual context-aware citation recommendation [28]. This is needed since we aim to perform the recommendation in an acceptable time. Differently to Gu et al. [28], we do not employ a faster local citation recommender as a prefetcher but a global citation recommender operating on the section-level. Altogether, our pipeline SCOPE (Section Citation Online PipelinE) entails four sub-tasks as depicted in Figure 1, namely structure analysis, cite-worthiness detection, a global citation recommender as the prefetcher, and a globally enhanced context-aware citation recommender. In the field of citation recommendation [22], a self-supervised learning approach is commonly adopted assuming that existing articles are citing correctly. We follow this self-supervised approach for all sub-tasks, except the structure analysis where we perform active wrapper induction.

In summary, the contributions made in this work are:

- We probe to what extent the sections of papers in the fields of artificial intelligence and machine learning can be mapped to generally applicable templates of how scientific papers are structured. By introducing wildcards, we were able to map 82% of the papers to one of our templates, though, it remains challenging to map to specific templates without any wildcards as the heterogeneity of paper structures is too high.
- We explore how state-of-the-art deep neural networks can be employed in an online pipeline for citation recommendation, using information from a paper that is currently being written. Our results show that the abstract of the citing paper is not required for the prediction, whereas removing the title of the citing paper from the input results in a drop in performance.
- We investigate how the individual modules in the pipeline improve when utilizing section information. While explicitly adding the section information to the input shows little effect, performing predictions on the section- and paragraph-level consistently leads to an improvement of the individual pipeline components.
- We evaluate the performance of different configurations of the pipeline. First, we experiment with the combination of prefetcher and local citation recommender. Afterwards, the complete pipeline is examined. These experiments emphasize the importance of employing a well-performing prefetcher and cite-worthiness detection module since they set an upper limit on the performance of the local citation recommender and hence the quality of the final output of the pipeline.

The remainder of the paper is organized as follows: Combined with a literature review, we reason our choice on how to implement the structure analysis and which deep neural networks to use for the other three sub-tasks in Section 2. Afterwards, we present our modular end-to-end online pipeline SCOPE for a global context-aware online citation recommendation in Section 3. Section 4 explains our experimental setup along with their results in Section 5. In Section 6, we discuss our results and give an outlook on future work, before we conclude in Section 7.

2 RELATED WORK

To start with, we brief the citing behaviour of scientists with regard to different sections. Then we look at the structural analysis of scientific papers. Afterwards, we review approaches to context-aware and global citation recommendation considering their applicability to our pipeline SCOPE. Finally, we give an overview over the field of cite-worthiness detection and its employment in citation recommendation systems. The section is concluded with a summary.

2.1 Analysis of Citing Behaviour

Among others, Bertin et al. [10] investigated the distribution of citations among the different sections of scientific articles. They concluded that the number of citations and the year of publication of the cited papers are correlated with the sections in the IMRaD [63] structure, which is short for *Introduction*, *Methods*, *Results* and *Discussion*. More precisely, most of the citations are located in the *Introduction* followed by the *Discussion* section and the recency of the cited papers increases from the *Methods* over the *Results* to the *Discussion* section. The age of citations in a scientific article can even be modeled by parameterized distributions such as the gamma distribution [64]. So far, these findings have not been integrated into approaches to citation recommendation, although they show potential to improve recommendation performance [22].

In our citation recommendation pipeline, we consider the structure of the citing paper on the section-level in all of our modules. Therewith, we aim at utilizing the above-mentioned correlations and thus enhancing the overall recommendation output.

2.2 Structure Analysis

A great deal of information can be extracted from the general structure of a document. We will focus only on the section headings, how we can extract them and what information they can give us about the document.

Wang et al. [73] performed an extraction of the argument structure, using a sequential pattern mining (SPM) algorithm, where they specifically examined the positions and the distribution of the argument units. In this process, the annotated scientific papers are transformed into a group of sequential tags. These tags can be mined in order to find recurring sequences from which the argumentation patterns can be identified. They observed that certain categories (Introduction, Related Work) tend to appear more frequently at the beginning, and others (Methods) tend to be a bridge between categories. They did this to study differences in structure between scientific fields, e.g. between biomedical research and information science. We limit our research to the field of computer science only, since there may be differences even within a field. Our goal is to extract the section headers and their positions, but chose to use the active wrapper approach instead.

One approach to learn rules by means of examples is active wrapper induction, which belongs to the broader field of information extraction. The wrapper obtains rules from examples and counterexamples by input of user feedback, which consists of which rules to learn and which to discard.

Recent work by Epp et al. [21] used active wrapper induction with their STEREO tool to find regular expressions for statistics extraction in scientific papers. They developed an approach centered on two sets of rules, R^+ and R^-, the former resembling rules that actually refer to statistics and the latter is the set of rules that confirms that a sentence does not contain any statistics. Our approach differs from this approach in one respect. We construct our own set of templates based on literature and expert opinion and match the items against them. We use the information obtained from the Active Wrapper to map the section headings to the common set of section titles we have defined.

In 1999, Muslea et al. [53] worked on an inductive algorithm called STALKER that generates extraction rules with high accuracy based on user-defined training examples. For this purpose, they have developed the embedded catalog (*EC*) formalism. This is a tree structure in which the leaves contain the information relevant to the user. With this method, all documents that are structured to a certain degree can be described, which is ideal for HTML documents.

One of the concepts to define the extraction rules are landmarks. These landmarks are used for efficient navigation because of their expressiveness. The so-called "linear landmark" consists of a sequence of tokens and wildcards. The words "Number", "Sign" and "HtmlTag" are typical tokens, because of their consistent occurrence in any HTML document. A wildcard could be any term that is related to an HTML document.

Crescenzi et al. [16] introduced the ROADRUNNER algorithm in 2001. Their approach for automated data extraction focuses on similarities and differences. ROADRUNNER was designed to be different from earlier active wrapper approaches. Gold [27] already made the assumption in 1967 that regular grammars cannot be correctly identified by positive examples alone. Probably, positive and negative examples are not sufficient either, but additional information, e. g., labels, must be present for an efficient identification. These generally shared some common features, as that the wrapper needs additional information, which is typically provided by the user and it is assumed that the wrapper has *a priori* information about the general structure of the schema.

2.3 Context-aware Citation Recommendation

A context-aware citation recommender [22] is required whenever one is searching for papers in order to back up specific citation contexts. The methods for this recommendation task differ in terms of the information utilized from the candidate papers and the citing document (besides the citation context, which is always included). We are interested in methods utilizing only meta-information for the candidate papers. This allows for a larger pool of candidates as the content of these is not required to be retrieved. For the citing paper, we generally allow any kind of information. However, since we do not want to personalize the recommendation, our approach remains agnostic with respect to the authors.

In general, any method from the field of information retrieval [50] can be applied to the task of local citation recommendation by viewing the citation context as the query and the candidate papers as the documents (e. g. [19, 30, 68]). In the following, we review recent context-aware citation recommenders and discuss their applicability to our approach. In order to improve the representation of candidate papers, one can utilize the k-hop neighborhood of the citation graph. In this context, Jeong et al. [34] proposed a BERT-GCN model in which the concatenation of context embeddings and citation graph embeddings is input into a feed-forward network. The former embeddings represent the citation context and are obtained from a BERT [18] model. The latter originate from a graph convolutional network (GCN) [38] using a citation graph containing the candidate papers as an input. Therefore, the citation relationship between the existing papers is utilized when representing a candidate paper. However, the construction of the citation graph requires processing the bibliographic information of all the candidate papers, which is not necessarily included in the meta-information. Furthermore, a recent survey by Ali et al. [1] assessed the BERT-GCN to be challenging to reproduce.

Similarly, the local information represented by the citation context can be enhanced by global information from the citing paper. This was implemented in the DualLCR model by Medić and Snajder [51], where the additional global information consists of the title and abstract of the respective paper. They employed a modular approach where different scores are combined to a final recommendation score by a non-linear layer. The two scores proposed by them are the semantic score and the bibliographic score. The former calculates the cosine similarity between the globally enhanced representation of the citation context and a candidate paper. Based on the pretrained Word2Vec embeddings from Bhagavatula et al. [11], these representations are learned by Bi-LSTM [32] cells followed by an additive attention [5] layer. The bibliographic score indicates the popularity of a candidate paper taking the names of the authors and how often the work got cited so far into account. DualLCR performed worse when utilizing the semantic score alone [51]. Nevertheless, we refrain from calculating a bibliographic score. In our view, this poses the risk of explicitly including undesirable citation biases [67] present in published papers and hence in the dataset created for self-supervised learning. These biases arise for instance from preferring popular papers and papers written by oneself or colleagues as a citation. Moreover, Medić and Snajder [51] achieved better results when globally enhancing the representation compared to using the text of the citation context only. We also utilize global information. However, we are not using the abstract of the citing document. The reasons for this are twofold: The citations within the text might change due to modifying the abstract since it is a major part of the input to the model. An even more severe circumstance is that the abstract might not exist yet in our scenario. In case of the title, one may assume that there is a working title available that will not be far from the final one.

The previously reviewed methods BERT-GCN and DualLCR are both outperformed by the SciBERT Reranker [28], which feeds the globally enhanced representation of the citation context and the candidate paper through a SciBERT [7] model followed by a feed-forward layer that outputs the relevance score. Therefore, we apply this model in our pipeline. Furthermore, Gu et al. [28] implemented the reranker in a two-stage telescope citation recommendation pipeline, i. e., they applied a fast prefetching model before the slower SciBERT Reranker. This reduces the amount of candidate papers that need to be processed by the reranker. In this way, the overall required computation time is improved, which is especially important for a real-time application of the pipeline in the light of the immense amount of papers that exist. Differently to previous works (e. g. [20, 44, 51]), they consider a "real" prefetcher for the evaluation. This means that the ground truth paper may not be included in the prefetched candidate list. Their proposed prefetching module is another local citation recommender named Hierarchical-Attention text encoder (HAtten), which efficiently computes embeddings for the citation context and the candidate papers and applies the cosine similarity in order to rank the candidates. Its recommendation only needs to be good enough such that the actually cited paper is contained in the top k (e. g. $k = 2,000$) ranked candidate papers, which are then passed on to the SciBERT Reranker. HAtten is able to outperform the hard-to-beat [28] BM25 [60] baseline in terms of both prefetching time and retrieval performance. Instead of a context-aware citation recommender, we employ a global citation recommender as the prefetching module. A local citation recommender needs to prefetch candidates for each citation context individually. In contrast, the global citation recommender only needs to be run once per citing document or, as in our case, once per section of the citing document. Due to the reduced amount of runs, we can use a slower prefetcher without harming the overall time required for prefetching candidate papers for a single document.

2.4 Global Citation Recommendation

Unlike local citation recommendation, global citation recommendation implies the task of suggesting citations without consideration of any local information like the citation context. It is therefore not context-sensitive and only uses the information on already known references for a given paper. This is similar to paper recommendation tasks. Therefore paper recommender can be generally adapted to the task of global citation recommendation. We look specifically for methods that do not create personalized recommendations to avoid eventual citation biases.

A recurring approach for this task is to use the citation graph of a given corpus of scientific papers to retrieve sets of citation candidates [13, 14, 17, 76]. Cai et al. [13] employ a model which simulates the citation network as a graph between papers and other heterogeneous objects like authors. The text associated with a paper, such as title and abstract, is encoded into a vector, which is forwarded to an energy-based adversarial network [78] and then used to learn the structure of the citation network.

Other work focuses on content-based recommendation by comparing the titles, abstracts and other global metadata of the different papers [12, 55, 71, 77]. Bhagavatula et al. [12] project documents into an embedding space and retrieve the k-nearest neighbours as citation candidates for a paper. Those candidates then get ranked according to their estimated probability of being cited for the current document. Using a prefetcher and a second, computationally more complex model is a pattern already deployed in global citation recommendation systems themselves. Nogueira et al. [55] use a BM25 prefetcher to retrieve the top k papers and forward them to a SciBERT reranker model [54] as query consisting of the titles and abstracts of the candidate papers.

Vagliano et al. [71] perform global citation recommendation using different autoencoder architectures on common datasets for this task. All autoencoder architectures encode the information which sets of papers get cited together in a candidate paper. In addition to that, they examined the influence of different metadata such as title, authors, year, and venue on the recommendation results. The results of the different encoder architectures heavily varies with the used dataset. So can a model which outperforms the others on one dataset seriously underperform on other datasets. Over all datasets, the adversarial autoencoder performs not always the best but with the most consistency.

Since our global citation model is only applied as prefetcher, we are not interested in the exact position of the positive candidates in the resulting pool but rather in the fact that it is included in the same. Therefore, we are able to compromise on recommender performance in exchange for better trainability and lower computational costs. We continue to use the work by Vagliano et al. [71] due to the available implementation and easier expandability for our experiments, but in general any global citation model can be applied for generating the pool of candidate papers for the local citation module.

2.5 Cite-worthiness Detection

Most works do not consider the application of context-aware citation recommendation [22] in a scenario like ours. They make use of the self-supervised dataset containing the exact positions of the citation markers to be predicted. Assuming that the given positions are correct, no additional cite-worthiness detection is performed. Consequently, the citation context is commonly defined by a certain amount of characters surrounding the given marker. For our scenario, however, the task of identifying cite-worthy citation contexts needs to be addressed as well.

A rather convenient option for cite-worthiness detection is to rely on the user to indicate the position of the citation markers as done in the CiteSight [44] system. He et al. [29] applied a windowing approach to the scientific document and experimented with four different methods in order to predict the cite-worthiness of each of these windows, namely, language models, contextual similarity, topical relevance, and dependency feature model. The topic distribution approach was applied practically in the RefSeer [33] recommender on sentence-level as this information was already available from the additional global citation recommendation they performed. A more detailed study on beneficial input to support vector machines (SVM) [31] and maximum entropy (ME) [9] classifiers for identifying cite-worthy sentences was performed by Sugiyama et al. [65]. They concluded that proper nouns and the context represented by the previous and next sentence are the most helpful features. Färber et al. [24] were able to outperform the SVM and ME classifiers in terms of precision by using a CRNN [42] architecture that combines convolutional with recurrent neural networks and applies custom word embeddings. The local citation recommendation system CITEWERTs [23] makes use of this CRNN for a sentence-wise cite-worthiness detection. Afterwards, the citation recommendation is performed by applying a latent semantic index on the noun phrases. They do not provide an evaluation of their approach as they are not aiming for state-of-the-art results but at making the research community aware of the importance of considering cite-worthiness detection as part of the recommendation pipeline.

Wright and Augenstein [75] introduced the dataset CiteWorth for cite-worthiness detection on the sentence-level. They verified the high quality of the dataset and showed that pretrained language models are able to achieve state-of-the-art results, outperforming previously presented methods. Furthermore, they already included contextual information by considering a whole paragraph when deciding on the cite-worthiness of a single sentence. Thus, we utilize their rules for creating a dataset along with their classification approach in our pipeline. As a consequence, we do not know the amount and position of citation markers in a cite-worthy sentence at inference time. Hence, we remain agnostic regarding this information during the training phase, as well.

2.6 Summary

There is a correlation between the sections in the IMRaD structure and the amount and recency of cited papers, which has so far not been exploited by citation recommendation research. In order to make use of this correlation, we perform a template-based structure analysis on the section-level with synonyms for section headings learned by means of active wrapper induction. Moreover, state-of-the-art methods for citation recommendation include deep neural networks for natural language processing. Yet, existing citation recommendation pipelines and systems [23, 29, 33, 44] do not make use of these networks. Instead, they apply methods closely related to the field of information retrieval. Employing deep neural networks in a citation recommendation pipeline poses the risk of impacting real-time performance. We approach this problem by utilizing a global citation recommender as a prefetcher. So far, only fast local citation recommenders have been employed as a prefetcher [7] and not in a citation recommendation pipeline taking a whole document as an input. Additionally, we have to adapt the input of the citation recommenders to our online scenario. This is due to the circumstance that the original implementations of the recommenders utilize global input data, such as abstract and title, that might not be available for a work-in-progress paper. Finally, it is noteworthy that we consider cite-worthiness detection as a part of the citation recommendation pipeline.

3 GLOBAL CONTEXT-AWARE CITATION RECOMMENDATION PIPELINE

In order to perform global context-aware citation recommendation in an online manner, we present the modular pipeline SCOPE depicted in Figure 1. The input is a LaTeX document of a paper that is currently being written. We first preprocess the document applying the results of our structure analysis. This leaves us with a bibliography, section headings fitting one of our templates, and the paragraphs of each section, split into sentences. Second, we identify citation contexts on the sentence-level with the CiteWorth [75] module and propose a pool of candidate papers for each section applying the AAE-Recommender [71] as a global citation recommender. In general, these two modules can work in parallel as their input is independent from each other. Finally, this information is input to the context-aware citation recommendation module, which is in our case the SciBERT Reranker [28]. The output is a ranked list of candidate papers, one for each identified citation context. The local citation recommender can start processing citation contexts as soon as the candidate papers for the first section are available. More detail on the single steps is given subsequently.

3.1 Structure Analysis

3.1.1 Pipeline. The structural analysis process is divided into a fixed sequence, which all documents traverse.

(1) Preprocessing / Lemmatizer
(2) Synonym Dictionary
(3) Duplicate Removal
(4) Template Matching

These 4 steps represent our pipeline for processing the documents to achieve an unified representation. First we extract the section headings. We then use a synonym dictionary to map the section headings to our section type set. Our section type set consists of the following terms:

- Introduction
- Related Work
- Method
- Experiment
- Result
- Discussion
- Conclusion

A section heading is defined as any possible heading which we can map to our section type set. A mapping might be as follows:

- Method ← Methodology
- Method ← Material and Method
- Method ← Procedure

All terms in this example are assigned to the "Method" section. The complete dictionary of synonyms is available in Appendix A.2.

Our synonym dictionary consists of the section headings mapped to the sections types we defined at the beginning of the experiments and those we learned through training. Then, all section headings are replaced by their section type counterpart. This can lead to duplicates, which we then remove. Duplicates occur when multiple sections, refer to the same section, such as in the original document the “Methods” section could be spread over three sections. After the mapping, however, there would be three times “Method”. We refer to the sequences of section headings resulting from this extraction as section sequences.

As the final step, we compare the section sequence prepared in this way with our templates. The templates consist of frequently used section sequences. For example [Introduction, Related Work, Method, Experiment, Discussion, Conclusion], is a frequently used and recommended section sequence. We iterate over our tree structure (see Figure 2) and count the number of matches a section sequence receives. Since there are parent-child relationships between templates, and by using wildcards, a section sequence can also match to multiple templates. During the training, we also learn new terms for our synonym dictionary. This step could also have been completely automated. However, for the pipeline we need terms for our synonym dictionary. We achieve this through an active wrapper approach. We manually go through all sequence headings and if a sequence sequence heading does not match the templates, we decide whether individual terms should be included in the dictionary.

3.1.2 Preprocessing. To extract the section headings from the LaTeX files, we use TexSoup. Using latexpand and then parsing with TexSoup caused errors in some documents. Some files no longer contained sections and others no longer contained meaningful headings. This may be due to the fact that some documents do not contain a meaningful structure or errors occur during preprocessing (Appendix A.1). Some cases of these parser errors are particularly common (see Section 6.2.2) and in these cases we can fix the errors. However, it was not possible to cover each of these edge cases within the scope of the project.

The results thus obtained are now in standardized form. As a subsequent step, we have included a lemmatizer, which brings cases such as “methods” and “method” to the same stem. This allows us to generalize many cases and accommodate small style differences.

3.1.3 Synonym Dictionary and Duplicate Removal. The next step in unifying the representation of the section sequence is the synonym dictionary. We exchange terms which have identical meanings. Due to different definitions by the authors of the documents, it is possible that sections, such as “Experiment” span multiple sections. Thus duplicates are created when the terms are exchanged by our synonym dictionary. We remove these duplicates and then match the section sequence with our templates.

3.1.4 Templates. We have chosen to represent the large number of heterogeneous section headings that occur in the section sequences by templates. Through literature review, we gained insight into different writing styles [2–4, 15, 25, 37, 40, 46–48, 57, 59, 69, 74]. Despite some commonly accepted principles for building a meaningful structure, anyone is free to create a structure as long as certain criteria are followed.

For example, it is welcomed to start with “Introduction” and end with “Conclusion”. “Related Work” should also come either at the beginning or at the end, and not in the middle. In addition, “Acknowledgement” and “Appendix” belong after “Conclusion”. However, this can make clear assignments difficult. An example of a reasonable structure would be a section with introductory words outlining the basic idea, followed by a section citing similar literature from the field.

We conducted a literature review to identify a variety of works on scientific writing, including style guides, papers, and slides that focus on common structures in the field of computer science. Based on these guidelines and incorporated expert knowledge, we decided to organize our templates into a tree structure, see Figure 2, since the tree enables us to model inclusion of different templates in an hierarchical manner. We do not claim completeness with our defined tree, but wanted to establish a hypothesis and test it based on our experiments. We have included two wildcard symbols in our template tree:

- “⋆” can be replaced by any number of sections, including none.
- “_” can be replaced by exactly one section.

The wildcards prove useful when subject-specific terms are used instead of common section names such as “Method”. The symbolic root for our tree is the star symbol, which contains all possible combinations of section sequences. We have included it only for the sake of completeness, since we process all documents. On the next level comes our actual root node [Introduction, ⋆, Conclusion]. This represents our most general condition, that every document starts with the Introduction and ends with the Conclusion. There may be an arbitrary number of sections in between.

On each level, our templates become more specific. The next level contains one "⋆" wildcard. On the second level we use "_" wildcard. The lowest level is the most specific. Wildcards are no longer used here. In almost all cases, we replace the "_' wildcard with "Method". The "Method" section has a specific method name in many cases. From the difference in how often our template matches "_" and "Method" we can infer the exact number.

3.2 Identification of Citation Contexts

Our implementation of the cite-worthiness detection on the sentence-level follows Wright and Augenstein [75]. They did not only introduce the CiteWorth dataset, but also investigate the performance of different methods in this task. Thus, we employ their best performing approach: a fine-tuned Longformer [8] model taking a complete paragraph of the document as an input and classifying each sentence within this paragraph regarding its cite-worthiness. To this end, each sentence in the paragraph is followed by a *[SEP]* token, which is a special separator token in language modelling introduced by Devlin et al. [18] for the BERT model. After applying the Longformer model, the representation of each *[SEP]* token is used in a feed-forward network for classification. This network consists of a single hidden layer with a tanh activation and dropout. The size of the hidden layer and the dropout rate are the same as in the Longformer model [75]. The Longformer model is fine-tuned end-to-end with the classification network by applying the cross entropy function as a loss.

We investigate three different ways of concatenating the section type from the structure analysis to the input. This leaves us with the following input variants, where "sec" and "sent" are short for section type and sentence respectively and *[CLS]* is a special classification token [18] for language modelling:

- *ctx*[1]: Does not contain any information about the section, i. e., *[CLS]* sent *[SEP]* sent *[SEP]* …
- *ctx-section-extra*: The section is in front of the input sequence followed by a *[SEP]* token, which is ignored in the classification network, i. e., *[CLS]* sec *[SEP]* sent *[SEP]* sent *[SEP]* …
- *ctx-section-first*: The section is in front of the first sentence in the paragraph, i. e., *[CLS]* sec sent *[SEP]* sent *[SEP]* …
- *ctx-section-always*: The section is in front of every sentence in the paragraph, i. e., *[CLS]* sec sent *[SEP]* sec sent *[SEP]* …

3.3 Creating a Pool of Candidate Papers

For this module, we use the work of Vagliano et al. [71] and adapt it to also consider metadata, which can be extracted from the LaTeX source code of a paper. The individual autoencoder models perform very differently on the different datasets, therefore we use the adversarial autoencoder, which provides the most consistent performance over all datasets. In the following, we refer to the model as AAE-Recommender. The adversarial autoencoder consists of three parts. First an encoder part, which translates the input to a hidden code layer, via a three-layer Multilayer Perceptron (MLP). Second, a decoder part, which also consists of a MLP with two hidden layers, which tries to recostruct the signals of the code layer to their original representation. To ensure that the distribution of the latent space of the code layer mimics a useful organization, the third part, the discriminator, tries to distinguish between the distribution of the latent space in the code layer and a prior selected distribution [49], i.e., a Gaussian distribution. The discriminator is a binary classifier that consists of the code layer as an input layer and one hidden layer. As an input the papers already cited by the citing paper are fed into the AAE-Recommender and the top k papers of the corpus are returned as a smaller candidate pool. Additional metadata information, which is known to contain meaningful words, meaning proper names and not nouns with a clear definitions, like the title of a paper are encoded using a TF-IDF weighted bag of words representation [26], with pretrained word embedding. For metadata information which does not contain meaningful words, like author names, a categorical embedding is trained from scratch, by randomly initializing embedding vectors for each value. Additionally, we investigate the influence on the recommendation performance of additional metadata, which we extract from our structural analysis of the papers. For this additional metadata, we use the section heading in which a paper gets cited and the citations divided by section. The encoded metadata is applied on the code layer of the autoencoder and therefore only used during the decoding step. We test the performance of the AEE-Recommender with different metadata configurations:

- None
- Paper Title
- Paper Title + Section Heading
- Section Heading

Moreover we test the performance using different aggregation policy on the dataset, first paper-wise by aggregating all cited paper per paper and second section-wise by aggregating all cited paper per section per paper.

A limitation which has to be considered is the fact that the AAE-Recommender can only recommend cited papers that are present in its training data split. A paper that was never observed in its training data also is not going to be retrieved during the evaluation. Especially for citation recommendation and the splitting based on different years, this becomes a limitation because scientific papers often cite contemporary paper in their fields. When the training split end serveral years before the queried papers, all newer papers which are not part of the training dataset are not going to be retrieved. Another limitation is the fact that all cited papers have to be known at training time, due to each cited paper being represented by a neuron in the output layer. When new cited papers should be included in the recommendations the model has to be retrained.

3.4 Performing the Citation Recommendation

This module performs the actual citation recommendation for each identified citation context with the respective pool of candidate papers. For this purpose, we apply the SciBERT Reranker [28] with slight modifications to the representation of the citation context. The SciBERT Reranker outputs a relevance score for each possible

[1] The *ctx* variant is adopted from the original work by Wright and Augenstein [75].

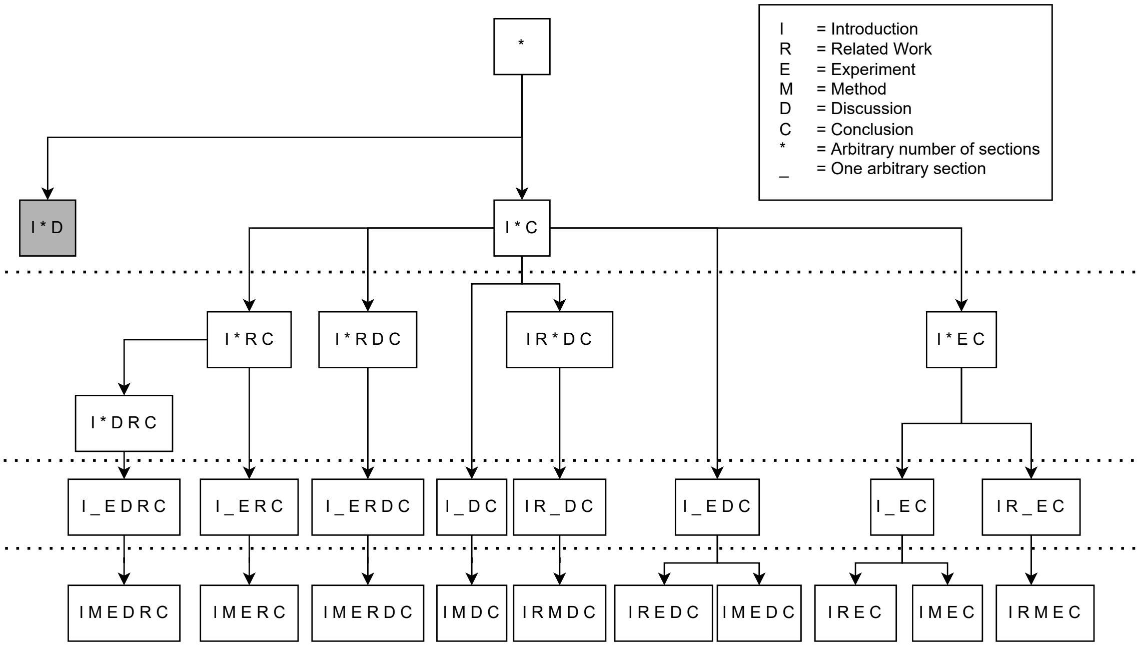

Figure 2: The abbreviations show the sequence of the section headings. An arrow from one template to another means that the parent template contains the child template. From top to bottom the tree becomes more specific, i.e. on the lowest level there are no more wildcards, but only unique assignments. Grayed out means that the template was added later, after some experiments.

combination of citation context and candidate paper by performing a regular binary-sequence classification task with the SciBERT [7] model into *relevant* and *non-relevant*, i. e., the input to the model has the following format [18]: *[CLS]* citation context *[SEP]* candidate paper *[SEP]*. The feed-forward network, subsequent to the SciBERT model, consists of a single linear layer and the relevance score is computed by applying the sigmoid function to the output for the *relevant* label [28]. Following Gu et al. [28] (and Medić and Snajder [51]), we apply the following triplet loss for fine-tuning the reranker:

$$\mathcal{L} = \max\{s(q, d_-) - s(q, d_+) + m, 0\} ,$$

where $s(q, d)$ is the relevance score for the citation context q with the relevant candidate paper d_+ or the non-relevant candidate paper d_-, respectively. The margin $m > 0$ defines the minimum score difference between relevant and non-relevant candidate papers.

Each candidate paper is represented by its title and abstract as in the original work. For the representation of the citation context, we extract the text of the sentence and the respective paragraph without any citation markers. Compared to the original work [7], we aim at using the paragraph of the citation context instead of the abstract of the citing paper and ignoring the title. We do this for two reasons: In our scenario, the abstract and the final title might not be available yet for recommending citations as the writing is still in progress. Furthermore, it seems unreasonable that recommendations for citations in the text change only due to altering the abstract or the title. Nevertheless, for purposes of comparison, we also perform experiments with the title and the abstract of the citing paper as an input. Optionally, we add the section type of the citation context (provided by the structure analysis) along with the year of publication of the candidate paper to the input.

Since the input, including our modifications, might be too long in order to be consumed by the SciBERT model (maximum input length: 512 tokens), we need to define a truncation strategy [66]. We make use of *longest-first*[2] truncation. This strategy removes token by token from the longer of the two inputs, i. e., the representation of the citation context or the candidate paper respectively, until the maximum input length is reached. Since the truncation is performed at the end of each representation, the order in which the information of the citation context and candidate papers is concatenated is relevant. When utilizing the paragraph in the representation, we aim to preserve the same amount of words before and after the citation context in order to account for equal importance of tokens before and after the sentence. For this reason, we implement a custom preprocessing strategy when the paragraph is utilized. Instead of truncating the paragraph at the end, we truncate it at the beginning and the end. In case there are not enough words before or after the citation context in the paragraph, we increase the amount of words on the respective other side of the context. Additionally, we introduce the term “TARGETSENT” and replace the sentence-based citation context in the paragraph with it. This has the benefit of reducing the amount of input tokens by removing redundant information from the paragraph since the citation context is already part of the input.

[2] https://huggingface.co/docs/transformers/pad_truncation

In summary, we represent the sentence-based citation context in the following ways for our experiments by concatenating the information with spaces:

- *default*[3]: sentence, title, and abstract
- accounting for the online scenario, where abstract and final title might not be available
    - *paragraph*: sentence, (working) title, and paragraph
    - *paragraph-notitle*: sentence and paragraph
- with section information
    - *section*: sentence, (working) title, section type, and abstract
    - *section-paragraph*: sentence, (working) title, section type, and paragraph
    - *section-paragraph-notitle*: sentence, section type, and paragraph

The candidate papers are represented by title and abstract in all experiments not containing the section type. When the section type is included, we additionally insert the numeric year of publication between title and abstract.

4 EXPERIMENTAL APPARATUS

In the following, we describe the used datasets along with their preprocessing for the individual components of the pipeline. After introducing different configurations of the pipeline and the employed baselines, we detail our evaluation procedure in a component-wise manner. This is followed by a description of our hyperparameter optimization and external sources utilized for our implementation. Finally, we give an overview over the measures applied during the evaluation.

4.1 Datasets

For the analysis of the whole pipeline and its individual components, different datasets are needed. All datasets contain papers written in English only. Across our arXiv and s2orc datasets, we split the papers to those published before or at 2018 and 2019 and later with the former serving as training data and the latter as test data. A split by year is often used in the application field of citation recommendation [52]. Also the same split across all dataset allows us to combine and evaluate different models together in our citation pipeline without having to worry about data imbalances.

4.1.1 arXiv. To analyze the structures of scientific papers, we use the original LaTeX code of each paper. Herefore, we utilize arXiv[4] because it publishes the LaTeX code of a paper next to a compiled pdf. We crawled the LaTeX source files for all scientific papers of arXiv following their rules for bulk data access[5] in the categories "artifical intelligence" and "machine learning", which have been released until the end of 2021. Per paper, we normalize all source files into a single LaTeX-file, using a script named "latexpand"[6]. Not all papers have their LaTeX code available. From all crawled 122,166 papers, 13,788 papers only have a compiled pdf available and for 24 documents the source code of the paper is a Microsoft Word file.

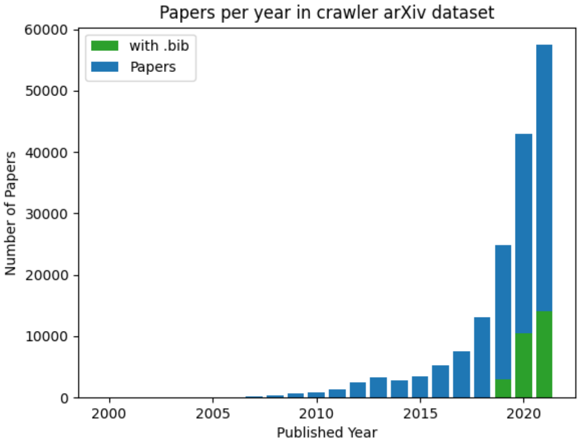

Figure 3: Distribution of Papers per Year. We also marked the share of Papers which have a *.bib* file available.

This leaves us with 108,354 papers which are used for our structural analysis. For the later use in our pipeline SCOPE, we also need a bibliography file to resolve citations. 27,615 papers have in addition to their LaTeX code, also their *.bib* file available. We deliberately ignore *.bbl* files due to their unpredictable structure. A distribution of the papers according to their publishing year can be seen in Figure 3. For the evaluation with the pipeline a paper must also be able to be loaded by TexSoup. This is the case for 3993 papers. The dataset for training contains all documents up to and including the publication year 2018. The remaining papers, published from 2019 to 2021, are the test dataset.

4.1.2 Modified S2ORC. For the evaluation of different configurations of our citation recommendation pipeline and to have comparable results between them, a dataset is needed that can be used for global and local citation recommendation as well as cite-worthiness detection. We use the general-purpose Semantic Scholar Open Research Corpus (S2ORC)[7] [45] as basis. The S2ORC dataset contains about 1.8 million fulltext papers across different domains. Given our proposed use case, we only consider papers that belong soley to the field of computer science. We then perform the following modifications. We remove all papers from the dataset that miss any entries required as an input to our modules later on in the pipeline. This leaves us with 455,785 papers. After that we apply the preprocessing of Wright and Augenstein [75] to extract a dataset containing potential citation contexts with masked citations. In addition, we also extract the section headings for each potential citation context in the dataset. This is needed for testing the performance using structurally enhanced metadata. We also extract the citations, which allows us to evaluate the performance for global citation recommendation. Finally, we match those citations to their respective citation context, which is needed for local citation recommendation.

The SciBERT-Reranker performs further filtering steps on the candidate papers by year such that citations into the future are precluded (see Section 4.2.4). Hence, we need to take care that

[3] The *default* representation is analogous to the one applied in the original work by Gu et al. [28].

[4] https://arxiv.org/

[5] https://arxiv.org/help/bulk_data

[6] https://gitlab.com/latexpand/latexpand

[7] https://github.com/allenai/s2orc

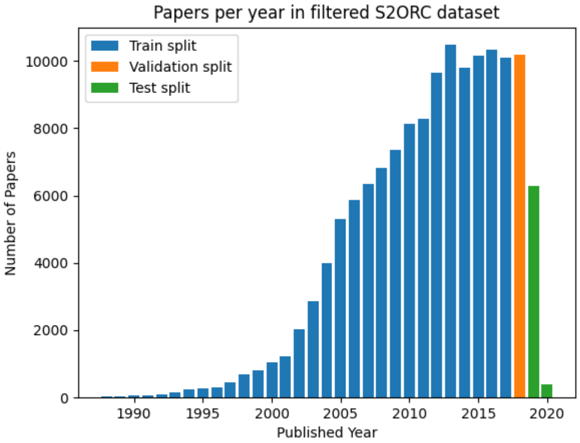

Figure 4: Distribution of Papers per Year. Only years with more than 50 papers are shown.

even after this additional filtering, at least k candidate papers are available for each citing paper. Therefore, we only consider citing papers published in or after the year where the cumulative sum of citing papers exceeds k. For $k = 2,000$ this corresponds to the year 2002. Candidate papers are all cited papers beginning with the year 1964, which is the first year of the S2ORC dataset. Because the annotation procedure of the S2ORC can sometimes allow citations into the future [45][8], we also remove citation links where the citing paper is older than the candidate paper. At last we remove all self-citations, meaning links where the citing and the cited paper are the same. After those steps we are left with $37,430$ citing and $27,065$ candidate papers. We split the dataset based on the publishing year of the papers. All citing papers published until the end of 2017 are contained in the train dataset, while the papers of the year 2018 form the validation dataset and papers from 2019 and 2020 are used as test dataset. A distribution of the papers according to their publishing year can be seen in Figure 4.

Based on the results of Bertin et al. [10], we analyzed our dataset regarding the frequency of citations in different sections and the relative citation age of these citations. The statistics are shown in Table 1. These findings show that there is a correlation between citation frequency and section as well as citation age and section. The most citations are found in the *related work* section while being rather absent in the sections which feature the own work of the authors like *experiment* and *method*. As for the age of the citations, it can be observed, that the citations which are included in the conclusion are rather recent compared to citations in the remaining parts of the papers.

4.1.3 ACL-200 [51]. With this dataset, we verify our reimplementation of the SciBERT Reranker by reproducing the results presented in the original work by Gu et al. [28]. Therefore, we apply the same split into training, validation, and testing data. This leaves us with $30,390$ training, $9,381$ validation, and $9,585$ testing contexts, and a total of $19,776$ papers published between 2009 and 2015 as the candidate papers.

Section	# sen	# c / sen	Age of Citation
Introduction	10.402	0.150	6.005
Related Work	11.239	0.122	5.827
Method	9.334	0.106	5.717
Experiment	10.927	0.079	5.106
Discussion	17.688	0.117	5.964
Conclusion	7.437	0.092	4.475

Table 1: Average age and frequency (# c(itations) / s(entence)) of citations in S2ORC dataset by sections. For better understanding also the average sentence count per section (# sen / section) is included. The section names got mapped by our predefined synonym list.

4.2 Preprocessing: Adapting the Datasets to the different Pipeline Modules

For our *Modified S2ORC* dataset, the section headings are mapped to the following set of section types by applying the synonym dictionary created by the structure analysis: *Introduction*, *Related Work*, *Method*, *Experiment*, *Discussion*, and *Conclusion*. Furthermore, we ignore the abstract as the amount of citations is negligible in this "section".

Below, we describe the preprocessing of the datasets for every module in SCOPE. Thereby, the resulting module-specific datasets follow the naming scheme *datasetName_moduleName*.

4.2.1 arXiv Dataset for the Synonym Dictionary. The TexSoup tool was used to extract the section sequences from the arXiv dataset. However, due to parser errors, not all documents could be processed. The remaining dataset contains 58,715 documents. From this, we created the training dataset and the testing dataset by splitting according to the year of publication as described above. This leaves us with 52,709 documents for training and 5,839 documents for testing. As shown in Figure 5, the dataset covers a wide variety of different lengths of scientific papers in regard to their section count.

4.2.2 ModifiedS2ORC_CiteWorth. For the cite-worthiness detection module, we further filter the paragraphs of citing papers in our *Modified S2ORC* dataset based on the position of the citation markers within the sentences. For the training and validation data, we only include paragraphs of the citing papers where every sentence is *CiteWorth-conform*, i. e., every sentence is conforming to the extraction rules defined for the CiteWorth [75] dataset. Especially, only sentences containing citation markers at the end are considered, meaning there are no inline citations. This way, the sentences stay grammatically correct and no spurious correlations are introduced by removing the marker [75]. For the test data, we create three variants: The first variant comprises all paragraphs of citing papers in the test split and is therefore named *test_all*. The second variant, called *test_conform*, consists only of paragraphs where every sentence is CiteWorth-conform. This is equivalent to the testing data in the CiteWorth dataset. The third variant is

[8] When a scientific paper gets updated after it was cited by another paper, it can happen that the corpus crawled the newer version of the paper, which now has a publishing date after that of its citing paper.

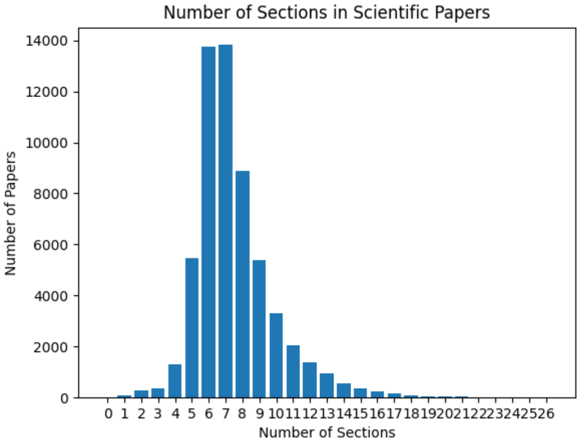

Figure 5: Number of sections in the arXiv dataset in computer science. In very rare cases, documents have more than 20 sections. The number of papers considered is 58,715. The average value for number of sections is 6.59

termed *test_non-conform*, containing only paragraphs where every sentence is not CiteWorth-conform. This enables us to investigate the generalizability of the cite-worthiness detection module trained with a CiteWorth-conform dataset to the wide variety of possible citation contexts, where the citation marker may be positioned arbitrarily within the sentence.

4.2.3 S2ORC_AAE-Recommender. In our *Modified S2ORC* dataset, each entry corresponds to one paragraph. This means, each paper is contained multiple times for each section of the paper, with the appropriate citation link. For the training and test data of the AAE-Recommender, we concatenate those entries such that each entry represents one section not one paragraph. To investigate the performance difference between training the AAE-Recommender section-wise and paper-wise, we also create a dataset where each entry corresponds to one paper and aggregate all citations for this paper.

4.2.4 ModifiedS2ORC_Reranker. For the reranker dataset, we consider the citation contexts of all citing papers in the *Modified S2ORC* dataset, i. e., all sentences that are cite-worthy and where the cited paper is contained in our dataset. Moreover, we only allow candidate papers that are published before or in the same year as the citing paper and exclude the citing paper itself. This causality constraint allows for a realistic training and testing scenario since it prevents citation into the future. This is especially important since we follow the common paradigm of *strict citation re-prediciton* [22] due to the self-supervised learning approach. This means that we only consider the paper cited by the authors as correct, while all other papers are viewed as wrong recommendations even though they may be relevant. Hence, our above introduced causality constraint counteracts the issue that a citation into the future might fit better than the paper cited back then.

4.2.5 ACL-200_Reranker. For this reranker dataset, we again want to apply our causality constraint introduced above in order to realistically filter out candidate papers. Since the year of publication is not given in the *ACL-200* dataset, we only consider the fact that a paper does not cite itself.

4.2.6 Complete pipeline. For a qualitative evaluation of all modules working together in our pipeline SCOPE as seen in Figure 1, we build a proof-of-concept setup. As dataset basis we use our self-collected and preprocessed arXiv dataset. Using TexSoup, we extract the needed structural information from the LaTeX- and bibliography files. For a given document, we extract

- the section headings present,
- the cited papers of a specific section, and
- the citation contexts of a specific sections, and mask their citation marker.
- for each citation context, the content of their citation marker.

To get the information of a cited paper in the LaTeXcode, we look up the entry in the bibliography file. We only match bibtex entries where the identifier matches exactly the citing key in the LaTeXsource code. Given this entry we use the title to find a corresponding match in our *Modified S2ORC* dataset. This means we remove every character that is not alpha numerical from the tile and then transform the title to lower case and look for an exact match. When a paper can not be found in our *Modified S2ORC* dataset this way, we map the entry to a paper with similar context by using BM25 on the preprocessed paper title and and the titles of all candidate papers in our *Modified S2ORC* dataset. The so extracted information gets forwarded to the remaining three models of our pipeline, namely the cite-worthiness Detection, the Prefetcher and the final Reranker, which then infer their respective recommendations for the user.

4.3 Methods

4.3.1 Configurations. First we evaluate each component of the citation pipeline on its own by analyzing the performance with our *Modified S2ORC* dataset. Also we evaluate the performance effects that added structural information, namely the section heading, achieves.

We then join the AAE-Recommender and SciBERT Reranker together and evaluate their combined performance by comparing it with the performance of several baselines (see Section 4.4.5). We use our *Modified S2ORC* dataset for this and employ the same test-split for all models.

We also combine all our four modules to a complete pipeline (as shown in Figure 1) and evaluate them in an online manner by using handpicked LaTeXsource files of our arXiv dataset (see Section 4.4.6).

4.3.2 Baselines. First, we introduce the *Synonym Dictionary Baseline*, which is used for verifying the need of defining a mapping of section headings to a common set of section types. We employ *Class-weighted Baselines* as an advanced chance baseline for cite-worthiness detection. The following three baselines, namely *Most Popular*, *Accumulated BM25*, and *Co-occurrence*, are applied for comparison with the global AAE-Recommender. Finally, *Local BM25* is a baseline for context-aware citation recommendation.

Synonym Dictionary Baseline. To test the necessity of the synonym dictionary, we calculate which support is achieved without exchanging terms. If no terms are exchanged, matching is significantly more challenging because the section titles occur in many small variations. This baseline shows the difference between no mapping at all and swapping the terms.

Class-weighted Baseline. This baseline performs a statistical analysis of the class occurrence in the training data. During the evaluation, it "throw a dice" with the properties of the analyzed data in order to come up with a label. Hence, this baseline is a more sophisticated version of a chance baseline by taking the actual class distribution into account. We make use of it as a baseline for the cite-worthiness detection module. Thereby, the section types can act as a prior for the label distribution.

Most Popular Baseline. The idea behind this baseline is the consideration that only a small portion of papers get cited way above average in comparison to the average paper [58]. The baseline exploits this correlation by counting how often a paper gets cited in the training data and uses this as a probability estimate for test data. We use the Most Popular baseline on the global citation recommendation prefetcher module.

Accumulated BM25 Baseline. The prefetcher module receives a list of already cited documents as an input. This baseline is based on BM25 [70], it processes the title of each already cited document to generate BM25 scores for all titles in the candidate pool and adding them for each candidate paper together. The accumulated scores are then used as a recommendation estimate.

Co-Occurrence Baseline. A strong baseline for global citation recommendation is the co-citation score [62]. The idea behind this baseline is the consideration that papers that have already been cited often together are more likely to be cited together again than papers which were less cited together. By constructing the co-occcurence matrix of all candidate papers in the training pool, we can obtain the co-citation scores by aggregating the co-occurence values of the already cited papers in the query.

Local BM25. This baseline performs context-aware citation recommendation by employing BM25 [60] as the recommender. The representation of the citation context with additional global information from the citing paper is the query and the candidate papers are the retrievable documents represented by their title and abstract. Ideally, the actually cited paper is getting the highest ranking score by BM25. The local BM25 can be employed in two ways: First, as a baseline for the reranker performing the actual citation recommendation. When calculating the BM25 weights for the k prefetched candidate papers, we consider the complete pool of potential candidate papers for the global weight terms. Second, as a prefetcher creating a pool of candidate papers for each citation context individually. It is not suited for a direct comparison with the AAE-Recommender since local BM25 is a context-aware citation recommender while the AAE-Recommender is a global citation recommender.

4.3.3 Oracle variants. In order to train and evaluate the SciBERT Reranker independently of the performance of the preceding modules, we introduce oracle versions for the cite-worthiness detection and prefetching module. When a component of the pipeline is applied in its oracle variant, it is performing its task flawlessly. For the cite-worthiness detection this means that all sentences are correctly classified regarding their cite-wothiness, i. e., all citation contexts are known. An oracle prefetcher is guaranteed to contain the cited paper in its retrieved list of the top k candidate papers. Hence, it cannot happen that the reranker does not have the chance to make a perfect ranking due to the cited paper being not part of its input candidate list.

4.4 Procedure

In general, we train our models only on the training split of the dataset. The validation set is only utilized to tune the hyperparameters of the modules. Even when evaluating on the test split, we do not perform any post-hoc training with the data in the validation split.

4.4.1 Structure Analysis. The structural analysis consists of two components: The template tree and the synonym dictionary. We match the section sequences to the template tree and calculate the support for each node. This information then flows into the citation recommendation.

Templates. As mentioned in section 3.1.2, we define templates, which are composed of frequently occurring section headings. These templates are arranged in a tree structure because there are relationships between them. The section sequences are iteratively compared with the tree. Then we count how many times each template has been matched.

We compute support values for section sequences that were not included in our template tree by adding a new template, namely [Introduction, ⋆, Discussion]. During manual training, we noticed that many section sequences end with "Discussion" instead of "Conclusion", but otherwise have a valid structure.

Synonym Dictionary. We randomly selected 500 documents from the dataset for training. The documents were manually iterated using the Active Wrapper approach and checked for synonyms. When a synonym is found, we map it to one of the terms in our shared set. After reviewing all 500 documents, we compute the support values for the templates. Afterwards, we conduct a second run in which the synonyms we learned are also considered in the matching process. For support measure, we use our test dataset of 5,839 documents. To measure how effective our dictionary is we calculate the support values without our synonym dictionary. This implies that the section sequences are matched to our template tree, but the terms are not replaced with our dictionary.

4.4.2 Identification of Citation Contexts. The cite-worthiness detection module is trained and evaluated with the *ModifiedS2ORC_CiteWorth* dataset. For each sentence in a paragraph, the CiteWorth module predicts whether it is cite-worthy or not. We then verify the prediction against the ground-truth label from the dataset. For comparison, we employ a change baseline and two class-weighted baselines, one with and the other without the section type as a prior.

4.4.3 AAE-Recommender. The test data is corrupted by removing a portion of cited papers, specified by the drop parameter d. Regarding this parameter, we use three different values to model different stages of the writing process of scientific work. Namely we use the values 0.2, 0.5 and 0.8 for our experiments. We evaluate the retrieval performance on our own S2ORC dataset. We also use this dataset to evaluate the performance-shift caused by additional metadata, namely the section heading.

4.4.4 Performing the Citation Recommendation. By the time we performed the experiments, there has been no implementation availalbe for the SciBERT Reranker yet. Hence, we first verify our reimplementation with the *ACL-200_Reranker* dataset, where the citation context is represented by the title and abstract of the citing document and the 200 characters before and after the citation marker. Afterwards, we conduct our experiments utilizing the *ModifiedS2ORC_Reranker* dataset.

The triplet loss is calculated per citation context with a single positive document and $n - a$ negative documents, where n is the overall batch size and a is the number of gradient accumulation steps applied. In order to reasonably reduce the amount of negative documents to sample from, we apply Local BM25 in its oracle variant for prefetching as in the original work [28] and only consider the top k ranked candidates. The oracle-BM25 represents the citation context by sentence, title, and abstract and the candidate papers by title and abstract, regardless of the input variant we apply to the reranker. For the training, we sample $n-a$ negative candidate papers. In the evaluation, we make use of all negative documents retrieved by the oracle-BM25. Later when employed in SCOPE, the prefetching is performed by the AAE-Recommender instead of the oracle-BM25.

Differently to previous works, we train and evaluate our context-aware citation recommender on the sentence-level without taking citation marker positions into account. Hence, we do not mask citation markers but simply remove them from the paragraph. Due to this, there may be multiple cited papers that need to be considered as positive documents. We deal with this circumstance differently in the training and the evaluation phase: During the training, we create an individual set of candidate papers for each cited document in the sentence. Every set only contains the respective cited document as a positive document, while the other cited papers are excluded from the candidate set. This way, we ensure that there exists only one positive document per batch while all other sampled papers are non-cited, negative documents. During the evaluation, we create a single set of candidate papers for each citation context. Every set contains all cited papers in the respective sentence as positive documents. Therewith, we allow more than one positive document and can validate whether all cited papers are ranked as high as possible.

We apply the Local BM25 without using the section information as a baseline for the experiments. Differently to the SciBERT Reranker, the input to the BM25 model is not restricted to a certain amount of input tokens. Hence, we do not perform any truncation and masking of the citation context in the paragraph with the term "TARGETSENT" is not required. BM25 is not suited for a comparison between the input variants with and without section information. The additional section and year information are not common between the representation of the citation context and the candidate papers. Due to this, BM25 de facto ignores the additional information. Thus, the performance is the same for both cases with section information and without it.

4.4.5 Prefetcher + Reranker. To evaluate the combined performance of our Prefetcher and Reranker module we join the bestperforming configurations of AAE-Recommender and SciBERT Reranker together and measure them on our *Modified S2ORC* dataset. We also compare them with several baselines, including global and local citation recommendation methods. For the Reranker we also test the performance of the Local BM25 baseline. For the Prefetcher module we employ the Co-Occurence and Most Popular baseline as a comparison for global citation recommendation.

The Local BM25 baseline is not an fair comparison to the AAE-Recommender due to being a local citation recommender that can employ additional information like the citation context, which will likely result in better recommendation performance. Nevertheless we use it for comparsion with the original work of Gu et al. [28]. We switch one neural network model against a baseline and test all combinations that arise this way.

4.4.6 Complete pipeline. We extract the structural information and citation context from the LaTeX file of a given paper as described in Section 4.2.6. For the recommendation we use the bestperforming configuration of each module. Because we can not evaluate the quality of a recommendation for a possible citation context which does not has a citation marker in the original paper, we do not perform a recommendation for those sentences. This also has the benefit of reducing the overall evaluation duration.

4.5 Hyperparameter Optimization

4.5.1 CiteWorth. The original work [75] performed a Bayesian grid search optimizing for the F1 metric on the validation data. Their CiteWorth dataset is based on the S2ORC dataset and we made use of their pre-processing steps for our *Modified S2ORC* dataset with additional filtering. Hence, we reuse their hyperparameters for the *Longformer-ctx* model independent of the applied input variant:

- batch size: 4
- number of epochs: 3
- learning rate: 0.00001112
- triangular learning rate warmup steps: 300
- weight decay: 0.0
- dropout probability: 0.1

4.5.2 AAE-Recommender. For the estimation of each parameter, we fixed the other to the values proposed by Vagliano et al. [71], and iterated over the respective search space. The search space for the hyperparameters of the is listed below:

- # hidden neurons: {40,60,80,100,120,140,160,180,200,220,240}
- # code neurons: {20,30,40,50,60,70,80}
- learning rate: {0.1,0.01,0.001,0.0001}

The number of epochs is estimated by using the validation split of our *Modified S2ORC* datasetand determined by which epoch the validation metrics (R@k for a fixed k) stopped improving. The the concluded bestperforming values where:

- # hidden neurons: 240

- # code neurons: 50
- learning rate: 0.001
- epochs: 20

The data threshold t, which indicates how often a paper must be cited before it is considered by the recommender, depends on the dataset and the memory available on the testing system. We set the threshold to 1 on our *Modified S2ORC* dataset.

4.5.3 SciBERT Reranker. The long runtime for evaluating the Reranker on the validation split of the dataset ($\sim$ 1.5 days on one NVIDIA A100 SXM4 40GB GPU) frames an extensive hyperparameter optimization infeasible. Since Gu et al. [28] apply the SciBERT Reranker to a further processed variant of the S2ORC dataset as well, we reuse their hyperparameter values:

- batch size n: 64
- gradient accumulation steps a: 2
- learning rate: 0.00001
- weight decay: 0.01
- margin m: 0.1
- amount of prefetched candidate papers k: $2,000$

The number of training epochs is not given by Gu et al. [28]. Thus, we investigate this ourselves by optimizing the mean reciprocal rank (MRR) with the validation data of the *ACL-200_Reranker* and the *ModifiedS2ORC_Reranker* dataset, respectively. For both datasets, we set the number of epochs to 5 after searching over {1, 2, 3, 4, 5} with the *default* input variant. More detailed results of the hyperparameter turning can be found in Appendix B.

4.5.4 Baselines.

Accumulated BM25. The ATIRE variant of BM25 contains three tuning parameters b, k_1, and ϵ. Without further investigation we set $b = 0.3$, $k_1 = 1.1$ and $\epsilon = 0.25$ as proposed in [70].

Local BM25. Without further tuning, we apply the default values $b = 0.75$, $k_1 = 1.5$, and $\epsilon = 0.25$. Medić and Snajder [51] conducted a grid search on the *ACL-600* dataset and found that those values of b and k_1 perform best in terms of R@10 and MRR.

4.5.5 Complete pipeline. We set the amount of papers k being prefetched to $2,000$ as this a commonly used value [28, 51]. With a larger k, the *R@k* metric is better while the computation time for the reranker increases. The threshold θ for classifying a sentence as cite-worthy is set to 0.5, following the evaluation setup in the original work [75].

4.6 Implementation

We compressed the datasets we downloaded from arXiv using the latexpand [9] tool. This was necessary because many documents were spread over several files. For the preprocessing of the structural analysis we implemented methods ourselves to extract information. For the definition of the templates, we consulted some guidelines, but then made our own selection. To parse the LaTeXdocuments and filter out the section headings, we use TexSoup[10].

For the AAE-Recommender as well as the CiteWorth module, we based our implementation upon the existing code provided by Vagliano et al. [71] and Wright and Augenstein [75], respectively. Changes were made in order to include additional metadata information and datasets. Since there has been no implementation available for the SciBERT Reranker yet, we reimplemented the reranker based on the simpletransformers[11] library.

4.7 Measures

Per experimental configuration, we introduce the calculated metrics. We start with the individual components, namely structure analysis, the CiteWorth module, the AAE-Recommender, and the SciBERT Reranker, followed by combining the prefetcher and the reranker, and finally the complete pipeline SCOPE.

For the structure analysis, we use wildcards in the construction of the tree, so it is possible for multiple nodes to match the same section sequence. Since we have defined the rules in the tree itself, we do not know if we may have forgotten important rules. If we calculate the support in the conventional way, then it would be difficult to see where we are still missing rules.

For the identification of the citation contexts, we apply typical metrics for classification with unbalanced datasets as in the original work by Wright and Augenstein [75], namely precision (P), recall (R), and the F1-score (F1). All three metrics are calculated in their binary variant with the cite-worthy sentences as the positive class.

For the AAE-Recommender, we calculate the macro-average of the rank-aware metrics reciprocal rank (RR) and recall@k (R@k) as in Vagliano et al. [71], where the RR is the fraction of one divided by the position of the first positive document. Differently to the original work, we are only interested in maximizing the R@k since it is only important that the actually cited paper is contained in the retrieved list of candidate papers, independent of the position. Note, the RR score of the AAE-Recommender has no influence on the overall pipeline performance since the ranking of the candidate papers is ultimately determined by the subsequent reranker.

For the SciBERT Reranker, we evaluate the mean R@k and mean reciprocal rank (MRR) scores like for the AAE Recommender above. This time, however, the MRR as well as the mean R@k are relevant since the output of the reranker is the list of recommendations presented to the user.

For the combinations of prefetcher and reranker, we make use of the evaluation metrics of the reranker, i.e., we calculate the mean R@k and the MRR scores for the reranker output.

For evaluating the complete pipeline, we cannot proceed the same way as for combining the prefetcher and the reranker since the pipeline might recommend citations for contexts that are actually not cite-worthy. In such cases, there exists no positive document, which is why the metrics R@k and RR are unsuitable for evaluation. Instead, we calculate the percentage of correctly treated citation-contexts as a metric, i.e., the accuracy of the pipeline is defined as

$$A@k = \frac{\text{number of correctly treated sentences}}{\text{number of sentences}} .$$

Citation-contexts are treated correctly when either of the following holds:

[9] https://gitlab.com/latexpand/latexpand

[10] https://github.com/alvinwan/TexSoup

[11] https://simpletransformers.ai/

- The sentence is not cite-worthy and classified as such by our pipeline.
- The sentence is cite-worthy and at least one of the actually cited papers (or a paper with similar context, in case the actually cited one is not in our *Modified S2ORC* dataset) is contained in the top k ranked papers recommended by our pipeline.

5 RESULTS

We present the results of our experiments broken down by component and pipeline configuration. Thereby, we follow the structure of Section 4.4.

5.1 Structure Analysis

The result of the experiments can be found in table 2. The left column shows the templates which the template tree (see Figure 2) consists of. The other columns contain the three experiment runs. The first experiment performed is performed without the synonym dictionary. Testing how many section sequences match the templates, if the section heading are not swapped with the section types. This requires an exact matching. On the first level a support of 47% is achieved. At the lower levels, however, support drops off considerably and only a few templates match. The second experiment swaps the sections headings for the sections types, which we defined at the start of the experiments. With the dataset, which consists of 52,709 documents, a support of 75% is achieved. On the lower levels, almost all templates match to some section sequences, but only two above 10%. We also use this step to learn new synonyms through our active wrapper approach. While manually iterating through the dataset, we check all section headings, which do not match any section types. If a term is synonymous with a section type, it is added to a list. This list will be added to the synonym dictionary, after the experiment concludes. The third experiment makes use the the section headings learned in the second experiment. The dataset used for this experiment consists of 6,006 documents. The extended synonym dictionary achieves a support of 77 %. The other templates all remained at a value below 10%. Variant [I * E C] achieves a support of 38% and one of its child nodes a value of 16%. These exceptions are:

- [Introduction, ⋆, Experiment, Conclusion]
- [Introduction, Related Work, _ ,Experiment, Conclusion]

Some templates we defined had 0 support. Among them are:

- [Introduction, _ , Experiment, Related Work, Discussion, Conclusion]
- [Introduction, Method, Experiment, Related Work, Discussion, Conclusion]
- [Introduction, _, Experiment, Related Work, Conclusion]
- [Introduction, Related Work, Method, Discussion, Conclusion]

The template tree we have defined is based on our review and expert opinion. We do not make any claim to completeness. The template tree has reached a support of 77 % at the first level. In addition, we get 5% through the second node in the first level [Introduction ⋆ Discussion] (see Figure 2). This increases our support for the first level to 82%, although we didn't use these nodes in the pipeline later on. This second node was added by us afterwards, because we noticed a constant 5% support for this node.

Table 2: The table shows the support for our templates. Thereby we calculate how often a list of section headers matched a template. 1) Support without SynDict 2) Support before training 3) Support after training

Template	1)	2)	3)
[I * C]	0.47	0.75	0.77
[I * D]	0.05	0.05	0.05
[I * R C]	0.05	0.06	0.07
[I * D R C]	0	0.02	0.02
[I R * D C]	0	0.05	0.06
[I * E C]	0.11	0.36	0.38
[I R _ E C]	0.08	0.13	0.16
[I R _ E D C]	0	0.03	0.04
[I _ D C]	0	0.01	0.02
[I _ E C]	0.02	0.05	0.07
[I _ E D C]	0	0.02	0.02
[I _ E R C]	0	0.02	0.02
[I _ E D R C]	0	0.01	0.02
[I _ E R D C]	0	0	0
[I M D C]	0	0.01	0.02
[I M E C]	0.01	0.03	0.03
[I M E D C]	0	0.02	0.02
[I M E D R C]	0	0.01	0.01
[I M E R D C]	0	0	0
[I M E R C]	0	0	0
[I R E C]	0	0.02	0.04
[I R M D C]	0	0	0
[I R M E C]	0.01	0.05	0.06
[I R E D C]	0	0.01	0.03

5.2 Identification of Citation Contexts with the CiteWorth Module

The results of the cite-worthiness detection experiments with the CiteWorth module are listed in Table 3. The Longformer models outperform the baselines irrespective of the input variant and the selected test data. The results for the *ctx* variant on the *test_conform* data are slightly worse than in the original work by Wright and Augenstein [75], where they achieved R=77.15%, P=59.92%, and F1=67.45% on their CiteWorth dataset with the same model variant. Comparing the results of the Longformer models among the different test datasets, the F1 metric is the highest for the *test_all* data and the lowest for the *test_conform* data. Compared to the *test_conform* data, the results on the other two test datasets show a decreased performance in recall, while the precision increases.

Adding the section type to the input, consistently improves the precision, while diminishing the recall of the CiteWorth module. The precision increases between 0.82% and 1.40% and the recall decreases between 0.98% and 1.73% for the three *ctx-section* variants compared to the *ctx* variant. For the *test_conform* and *test_non-conform* data, the F1 metric improves with section information, whereas it stays the same or even deteriorates for the *test_all* data. The class-weighted baseline cannot improve more than 0.95% in any of the three calculated metrics by adding the section type as a prior. However, the addition of the section type leads to an improvement of both precision and recall for the baseline.

5.3 Retrieving Candidate papers with the AAE-Recommender

We use the AAE-Recommender as a prefetcher. Thus we focus on the performance of the recall@k where $k = 2,000$ when evaluating the module on our *Modified S2ORC* dataset. We test different amounts of structural metadata as additional data in the decoding step. The results are shown in Table 4. The best-performing variant of the AAE-Recommender in terms of R@2000 is the one which uses the paper title and section headings as additional metadata in the decoding step, while being closely followed by the other variants. The best-performing baseline is the Most Popular baseline, which achieves comparable performance to the AAE-Recommender variants. It is followed by the Countbased baseline, which degrades its performance with higher drop parameters due to being dependent on the co-occurrence matrix which yields less information the more papers are removed. We also investigated the performance difference when using a paper-wise aggregated dataset and a section-wise aggregated dataset (see Section 4.2.3). The results can be seen in Table 5. The section-wise aggregation outperforms the paper-wise aggregation (as described in Section 4.2.3) by a considerable margin over all metadata configurations.

5.4 Performing the Citation Recommendation with the SciBERT Reranker

Our re-implementation of the SciBERT Reranker achieved an MMR of 0.547 and an R@10 of 0.768 on the *ACL200_Reranker* dataset. Comparing to the original work by Gu et al. [28], where they reported MRR=0.531 and R@10=0.779, we can conclude that their results are reproducible and that our re-implementation of the SciBERT Reranker is valid.

Table 6 presents the results on the *ModifiedS2ORC_Reranker* dataset for different input variants to the Reranker and the Local BM25 baseline. Independent of the input variant, the SciBERT Reranker is performing better than Local BM25.

The baseline achieves the best results in terms of both MRR and R@10 when utilizing the *paragraph* input variant without masking. In general, the *paragraph* variants of Local BM25 perform better than the *default* one. When removing the title of the citing paper, the performance drops by 0.020 on average compared to the respective input variants with title. The variants without masking consistently outperform their masked counterpart. By not masking the citation context, it is assigned more weight to by the BM25 algorithm. In summary, when performing a local citation recommendation with BM25, the title of the citing paper should not be removed from the input and the paragraph along with the sentence-based citation context itself are better suited than the abstract.

We can observe the same trend for the SciBERT Reranker: The best performance with and without section information is achieved when utilizing the citation context, the masked paragraph, and the title of the citing paper for representing the citation context. Replacing the abstract with the masked paragraph leads to a performance gain of 0.023 on average, while removing the title decreases the performance by 0.010 on average. Due to the attention mechanism [72], the SciBERT model itself can learn to assign more weight to the words in the citation context when performing the recommendation. Adding the section information explicitly, however, did not improve but rather degrade the performance of the reranker, especially when utilizing the input variant with the abstract. In general, one would not expect the SciBERT Reranker to drop in performance when adding additional information to the input since the attention mechanism [72] could learn to ignore these information in case they are not helpful for the prediction. Hence, this is a surprising result that we discuss among others in Section 6.1.4.

5.5 Prefetcher + Reranker

We test all reranker and prefetcher combinations when using our models and baselines as described in Section 4.4.5. It can be seen that the best prefetcher over all combinations is the local citation recommender Local BM25. From our global citation recommender, the best prefetcher is the Co-occurence baseline, followed by the Most Popular baseline and the AAE-Recommender. When comparing the result to 7 it can be seen as expected that the recall performance is limited by the R@2000 performance of the prefetcher. Also the AAE-Recommender, which is the bestperforming model in our prefetcher results, now is the least performant configuration. Also the Most Popular baseline is now surpassed by the Co-occurence baseline, that achieved a considerable worse performance in our prefetcher evaluation. Also note how an increased drop parameter leads to a worse performance with the Co-occurence baseline, which matches the results of the prefetcher evaluation, while it does not lead to any performance differences using the AAE-Recommender. The performance of the SciBERT Reranker with the best performing prefetcher is worse than its usage with the oracle Local BM25 (see Table 6). The performance of the Local BM25 reranker is only slightly worse than in its usage with the oracle prefetcher.

5.6 Complete Pipeline

We run our complete pipeline configuration over all citation context, that were extracted from the test-split of our arXiv dataset. The results can be seen in Table 8. For the prefetcher & reranker the R@5 and R@10 do not correspond to actual recall@k values because the amount of correct citations in the first k entries get not divided by the amount of all correct citations. Meaning only one entry has to be present that a citation context gets marked as correctly processed. Also only the values for prefetcher & reranker are only calculated for correct cite-worthy context, because for non-correctly classified context we can not perform a meaningful evaluation. From the total of $491,137$ tested citation context contain $66,879$ a citation and are therefore cite-worthy. An evaluation with the pipeline takes on average 110 milliseconds for the extraction (per sentence) and 100 milliseconds for the cite-worthiness detection (per paragraph). If a sentence is then classified as cite-worthy the pipeline takes another 42 milliseconds (per section) for the prefetcher and $5 * k$ milliseconds (per sentence) for the reranker module, where k is the amount of candidate papers prefetched. The timing was measured using a single NVIDIA A100 SXM4 40GB GPU and AMD EPYC 7302 CPU. There are on average 196.5 sentences and about 27.3 cite-worthy sentences in a paper.

Table 3: Cite-worthiness detection performance on the three test splits of the *ModifiedS2ORC_CiteWorth* dataset. For all metrics, we report the mean and standard error (value in brackets) over five different seeds in percent.

Model with input variant	**test_conform**			**test_non-conform**			**test_all**		
	R	P	F1	R	P	F1	R	P	F1
ctx	$\mathbf{75.85}_{(\mathbf{0.74})}$	$57.68_{(0.58)}$	$65.50_{(0.15)}$	$\mathbf{70.03}_{(\mathbf{1.04})}$	$62.60_{(0.66)}$	$66.07_{(0.25)}$	$\mathbf{70.24}_{(\mathbf{0.99})}$	$66.97_{(0.55)}$	$\mathbf{68.54}_{(\mathbf{0.25})}$
ctx-section-extra	$74.63_{(0.58)}$	$58.80_{(0.52)}$	$\mathbf{65.76}_{(\mathbf{0.12})}$	$69.05_{(0.51)}$	$63.61_{(0.71)}$	$66.20_{(0.25)}$	$68.94_{(0.42)}$	$67.89_{(0.52)}$	$68.39_{(0.12)}$
ctx-section-always	$74.12_{(0.68)}$	$58.50_{(0.41)}$	$65.37_{(0.08)}$	$68.85_{(0.67)}$	$64.31_{(0.30)}$	$\mathbf{66.49}_{(\mathbf{0.19})}$	$68.84_{(0.60)}$	$\mathbf{68.26}_{(\mathbf{0.30})}$	$\mathbf{68.54}_{(\mathbf{0.16})}$
ctx-section-first	$74.31_{(0.45)}$	$\mathbf{58.97}_{(\mathbf{0.29})}$	$65.75_{(0.09)}$	$68.87_{(0.60)}$	$\mathbf{64.00}_{(\mathbf{0.43})}$	$66.33_{(0.09)}$	$68.71_{(0.45)}$	$68.20_{(0.27)}$	$68.45_{(0.12)}$
chance baseline[a]	$49.97_{(0.49)}$	$28.40_{(0.24)}$	$36.22_{(0.32)}$	$49.81_{(0.27)}$	$30.84_{(0.14)}$	$38.09_{(0.18)}$	$50.01_{(0.24)}$	$35.55_{(0.13)}$	$41.56_{(0.17)}$
class-weighted baseline[b]	$26.73_{(0.41)}$	$28.38_{(0.36)}$	$27.53_{(0.39)}$	$26.68_{(0.15)}$	$30.85_{(0.15)}$	$28.62_{(0.15)}$	$26.83_{(0.15)}$	$35.61_{(0.16)}$	$30.60_{(0.15)}$
class-weighted baseline + section[c]	$27.19_{(0.44)}$	$29.20_{(0.29)}$	$28.16_{(0.37)}$	$27.31_{(0.08)}$	$31.66_{(0.10)}$	$29.32_{(0.09)}$	$27.71_{(0.16)}$	$36.51_{(0.18)}$	$31.51_{(0.17)}$

Values of ϕ determined by the baselines. ϕ of the times, a sentence is classified as cite-worthy.

[a] $\phi = 50\%$, [b] $\phi \approx 26.76\%$, [c] $\phi(\text{introduction}) \approx 27.74$, $\phi(\text{related work}) \approx 30.72\%$, $\phi(\text{method}) \approx 19.92\%$, $\phi(\text{experiment}) \approx 17.31\%$, $\phi(\text{discussion}) \approx 20.68\%$, $\phi(\text{conclusion}) \approx 21.77\%$

Table 4: Performance of Prefetcher on our *Modified S2ORC* dataset. Datathreshold was 1. For both metrics, we report the mean and standard deviation over 4,199 test records.

	$d = 0.2$		$d = 0.5$		$d = 0.8$	
	MRR	R@2000	MRR	R@2000	MRR	R@2000
AAE-none	$0.012_{(0.090)}$	$0.343_{(0.092)}$	$0.029_{(0.133)}$	$0.335_{(0.435)}$	$0.045_{(0.166)}$	$0.327_{(0.420)}$
AAE-title	$0.012_{(0.092)}$	$0.360_{(0.460)}$	$0.028_{(0.135)}$	$0.350_{(0.440)}$	$0.043_{(0.165)}$	$0.340_{(0.424)}$
AAE-section-title	$0.012_{(0.087)}$	$0.377_{(0.466)}$	$0.028_{(0.131)}$	$0.360_{(0.443)}$	$0.044_{(0.162)}$	$0.348_{(0.427)}$
AAE-section	$0.012_{(0.08)}$	$0.364_{(0.462)}$	$0.029_{(0.129)}$	$0.349_{(0.440)}$	$0.045_{(0.159)}$	$0.343_{(0.425)}$
Random baseline	$0.000_{(0.001)}$	$0.070_{(0.247)}$	$0.000_{(0.016)}$	$0.076_{(0.243)}$	$0.000_{(0.016)}$	$0.073_{(0.229)}$
Accumulated BM25 baseline	$0.010_{(0.082)}$	$0.206_{(0.388)}$	$0.018_{(0.112)}$	$0.144_{(0.319)}$	$0.011_{(0.087)}$	$0.077_{(0.234)}$
Most Popular baseline	$0.011_{(0.078)}$	$0.361_{(0.462)}$	$0.026_{(0.116)}$	$0.346_{(0.440)}$	$0.040_{(0.138)}$	$0.339_{(0.424)}$
Co-Occurence baseline	$0.033_{(0.159)}$	$0.243_{(0.414)}$	$0.053_{(0.200)}$	$0.163_{(0.339)}$	$0.033_{(0.161)}$	$0.083_{(0.245)}$

Table 5: Performance difference of Prefetcher on our *Modified S2ORC* dataset with paper-wise entries and with section-wise entries. Datathreshold was 1. For both metrics, we report the mean and standard deviation over 4,199 (sectionwise) and 3195 (paperwise) test records.

	$d = 0.2$		$d = 0.5$		$d = 0.8$	
	MRR	R@2000	MRR	R@2000	MRR	R@2000
AAE-none (paperwise)	$0.009_{(0.070)}$	$0.286_{(0.434)}$	$0.022_{(0.105)}$	$0.288_{(0.420)}$	$0.033_{(0.128)}$	$0.288_{(0.406)}$
AAE-title (paperwise)	$0.009_{(0.070)}$	$0.302_{(0.443)}$	$0.021_{(0.105)}$	$0.299_{(0.425)}$	$0.032_{(0.128)}$	$0.296_{(0.409)}$
AAE-section-title (paperwise)	$0.009_{(0.069)}$	$0.307_{(0.44)}$	$0.021_{(0.106)}$	$0.295_{(0.420)}$	$0.033_{(0.129)}$	$0.287_{(0.404)}$
AAE-section (paperwise)	$0.008_{(0.067)}$	$0.264_{(0.424)}$	$0.020_{(0.104)}$	$0.267_{(0.409)}$	$0.031_{(0.125)}$	$0.266_{(0.396)}$
AAE-none (sectionwise)	$0.012_{(0.090)}$	$0.343_{(0.092)}$	$0.029_{(0.133)}$	$0.335_{(0.435)}$	$0.045_{(0.166)}$	$0.327_{(0.420)}$
AAE-title (sectionwise)	$0.012_{(0.092)}$	$0.360_{(0.460)}$	$0.028_{(0.135)}$	$0.350_{(0.440)}$	$0.043_{(0.165)}$	$0.340_{(0.424)}$
AAE-section-title (sectionwise)	$0.012_{(0.087)}$	$0.377_{(0.466)}$	$0.028_{(0.131)}$	$0.360_{(0.443)}$	$0.044_{(0.162)}$	$0.348_{(0.427)}$
AAE-section (sectionwise)	$0.012_{(0.08)}$	$0.364_{(0.462)}$	$0.029_{(0.129)}$	$0.349_{(0.440)}$	$0.045_{(0.159)}$	$0.343_{(0.425)}$

When comparing the performance of the cite-worthiness module with its performance on *Modified S2ORC* dataset we notice that the recall as well as the precision metric are much lower than on the latter. This is also the case when comparing the performance of the prefetcher & reranker combination with its experiments on *Modified S2ORC* dataset. Here the R@10 value is about factor ten worse than on the other dataset.

The heavy class imbalance of cite-worthy and non cite-worthy citation contexts may be an explanation for the still high accuracy of the pipeline for all citation contexts achieve a value of 56.86 therefore treating more than half of the citation contexts correctly.

6 DISCUSSION

Given our results, we now examine how this translates to our research questions. We also reflect on where our experiments may degenerate in quality due to our constraints and the work which is needed to generalize to other domains. We propose additional research that can be conducted using our work as a basis.

Table 6: Citation recommendation performance on the test split of the *ModifiedS2ORC_Reranker* dataset, with the oracle variant of Local BM25 as the prefetcher and $k = 2,000$. For both metrics, we report the mean and standard error (value in brackets) over the 6,978 citation contexts.

Model with input variant	MRR	R@10
SciBERT-default	$0.606_{(0.005)}$	$0.785_{(0.005)}$
SciBERT-paragraph	$\mathbf{0.619_{(0.005)}}$	$\mathbf{0.810_{(0.004)}}$
SciBERT-paragraph-notitle	$0.611_{(0.005)}$	$0.796_{(0.005)}$
SciBERT-section	$0.594_{(0.005)}$	$0.778_{(0.005)}$
SciBERT-section-paragraph	$\mathbf{0.618_{(0.005)}}$	$\mathbf{0.808_{(0.004)}}$
SciBERT-section-paragraph-notitle	$0.608_{(0.005)}$	$0.790_{(0.005)}$
LocalBM25-default	$0.220_{(0.004)}$	$0.328_{(0.005)}$
LocalBM25-paragraph (with masking)	$0.237_{(0.004)}$	$0.336_{(0.005)}$
LocalBM25-paragraph (no masking)	$\mathbf{0.254_{(0.004)}}$	$\mathbf{0.357_{(0.006)}}$
LocalBM25-paragraph-notitle (with masking)	$0.217_{(0.004)}$	$0.311_{(0.005)}$
LocalBM25-paragraph-notitle (no masking)	$0.239_{(0.004)}$	$0.332_{(0.005)}$

Table 7: Citation recommendation performance on the test split of the *ModifiedS2ORC_Reranker* dataset, with different (non-oracle) prefetchers and $k = 2,000$. For MRR and R@10, we report the mean and standard error (value in brackets) over the 6,978 citation contexts.

Prefetcher		Reranker			
		SciBERT-paragraph		LocalBM25-paragraph (no masking)	
		MRR	R@10	MRR	R@10
$d = 0.2$	AAE	$0.235_{(0.005)}$	$0.294_{(0.005)}$	$0.098_{(0.003)}$	$0.139_{(0.004)}$
	Co-occurrence baseline	$0.336_{(0.005)}$	$0.432_{(0.006)}$	$0.176_{(0.004)}$	$0.240_{(0.005)}$
$d = 0.5$	AAE	$0.235_{(0.005)}$	$0.294_{(0.005)}$	$0.098_{(0.003)}$	$0.140_{(0.004)}$
	Co-occurrence baseline	$0.325_{(0.005)}$	$0.418_{(0.006)}$	$0.169_{(0.004)}$	$0.230_{(0.005)}$
$d = 0.8$	AAE	$0.235_{(0.005)}$	$0.294_{(0.005)}$	$0.098_{(0.003)}$	$0.139_{(0.004)}$
	Co-occurrence baseline	$0.308_{(0.005)}$	$0.396_{(0.006)}$	$0.158_{(0.004)}$	$0.216_{(0.005)}$
	Most Popular baseline	$0.239_{(0.005)}$	$0.303_{(0.005)}$	$0.111_{(0.003)}$	$0.153_{(0.004)}$
	LocalBM25-paragraph (no masking)	$0.512_{(0.005)}$	$0.665_{(0.005)}$	$0.231_{(0.004)}$	$0.327_{(0.005)}$

Table 8: Citation recommendation performance of the complete pipeline on the test split of our arXiv dataset. From the total of 491,137 citation context contain 66,879 a citation marker.

	Cite-worthiness Detection				Prefetcher + Reranker		Complete Pipeline	
	P	R	F1	A	R@5	R@10	A@5	A@10
Pipeline	18.44	49.01	26.75	63.53	4.30	4.97	56.86	56.86

6.1 Main Results

6.1.1 Structure Analysis in Computer Science Papers. The results were in line with our expectations. As previously mentioned in Table 2, we achieve a support of 82% on the first level of the tree. This confirms our assumption that the basic structure is represented by the first level. The first level in this case is the topmost (see Figure 2).

As shown in Table 2, the support values decrease at the lower levels and, with two exceptions, remain below 10 %. This result was expected by us, since it was already obvious during the creation of the tree that a multitude of different structures exist. The two exceptions came as a surprise, as we did not expect these two sequences to be used this often. We assume that with improved sequence recognition, e.g. Natural Language Processing, even more sequences can be matched to our tree.

6.1.2 Identification of Citation Contexts. Our results show that the CiteWorth module trained on *CiteWorth-conform* citation contexts, i. e., citation contexts that do not entail any inline citations, generalizes well in terms of F1 to the wide variety of citation contexts.

While the recall is lower for inline citations, the precision is higher. A possible explanation for the latter is that inline citations are most likely more often introduced by *signal phrases* than citations at the end of a sentence. Signal phrases are any collection of words that indicate a citation, such as "showed", "found", or "use". Hence, the network can utilize these signal words and perform a more precise classification, which on the other hand impacts the recall performance.

For the class-weighted baseline, one can see that adding the section type as a prior has an influence on the model, i. e., the proportion of cite-worthy sentences varies from section to section. Since the distribution of the purpose of the citations varies per section [36] and different citations utilize different signal phrases, the Longformer model could learn different signal phrase distributions per section. Due to this, the Longformer model should, in theory, be able to improve more than the baseline when adding section information. Our results support this theory in terms of precision. Nevertheless, the improvement of precision is not able to compensate for the loss in recall on the *test_all* data.

Finally, there is no model variant that is clearly performing best over all test data variants and metrics. The *test_all* data is closest to real-world data as it contains inline as well as citations at the end of sentences. Hence, the performance on this dataset is the most relevant for a later application. In a later application of SCOPE, the user is presented with recommendations and can decide by themselves whether they agree that a sentence requires a citation. Since the user can decide against the cite-worthiness of sentences, it is more important that the recommended list of cite-worthy sentences is complete than precise. Following this reasoning, we assume that a high recall is preferable. Even though, the precision should not be underestimated as it is not user-friendly to recommend citations for too many sentences that are actually not cite-worthy. Additionally, a high precision saves the computational effort for the contexts that would be rejected by the user anyways. On the *test_all* data, the highest F1 score is achieved by the *ctx* and the *ctx-section-always* variant. Moreover, the *ctx* input variant has the highest recall but at the same time the lowest precision on this test data. Overall, the *ctx-section-always* variant of the CiteWorth module serves our needs the best with having a high recall, precision and F1 on the *test_all* data.

6.1.3 *AAE-Recommender.*

The explicit addition of the section title to our training data did not improve the performance of the AAE-Recommender. Also the addition of the title as additional metadata did not provide a performance improvement regarding the R@2000 metric. In fact the best-performing variant on our *Modified S2ORC* dataset is the one which does not use additional metadata in the decoding step. This effect may seem counterintuitive for a citation recommendation task, but this is inline with the experiments conducted by Vagliano et al. [71] on the PubMed and ACM datasets with regards to the drop parameter d. The additional metadata does not increase the MRR of the AAE-Recommender on those datasets. This coincides with our results where the MRR does not change and only the R@2000 differs slightly. While the explicit additional information of which section the papers are cited in yielded no performance improvement, the grouping of citations by section rather than by paper, therefore implicitly adding the section information, improved the performance of the AAE-Recommender by a wide margin. There are multiple explanations for this effect. One the one hand grouping cited papers together in smaller sets would simplify the decoding from the latent space. One the other hand this also indirectly generates more data-points which can be used for training. Therefore, our hypothesis, that the addition of section information helps the performance of citation recommendation, holds true for our prefetcher module.

6.1.4 *Performing the Citation Recommendation.*

Previous work utilized the *default* input variant when employing Local BM25 as a baseline or as a prefetcher (e. g. [28, 51]). Our results, however, suggest to replace the abstract with the paragraph of the citation context. Also the SciBERT Reranker performs better with the paragraph than with the abstract as an input. Hence, the paragraph-level input might be a better choice for performing context-aware citation recommendation in general, especially when the citation context is a single sentence and not a larger window around the citation marker.

An unexpected finding is that the performance of the SciBERT Reranker drops when adding the section type and the year of publication of the candidate paper to the input. The existing correlation between section and recency of the cited papers as investigated by Bertin et al. [10] could not be exploited by the reranker. Nevertheless, the preformance should not drop due to the additional information since the SciBERT model has the capability of ignoring these information by not attending to it [72]. In the following, we discuss three possible explanations for this observation: First, the additional information consumes input tokens, which potentially leads to tokens in the abstract or paragraph to be removed from the input. However, the amount of removed tokens is negligibly small and this is only an explanation for inputs that exceed the maximum input length of the SciBERT model. From this reasoning it would follow that the textual information is more relevant than the explicit notion of the section and the recency of candidate papers. Second, we did not perform hyperparameter tuning for an input variant with section information and the additional information might have increased the sample complexity for the learning problem. Finally, we only ran the experiments for one seed. Thus, more extensive experiments might not give rise to this unexpected behaviour.

6.1.5 *Prefetcher + Reranker.*

As expected the R@10 performance of our Prefetcher + Reranker combinations is highly correlated with the R@2000 performance of the prefetcher module. This is seen the fact that the bestperforming LocalBM25 prefetcher outperforms our global citation recommendation prefetchers by a large margin. This is rooted in the fact that this prefetcher achieves a R@2000 of 0.798 on the dataset.

An unexpected finding is the fact that the performance of the global citation prefetcher differs highly from their performance in our prefetcher evaluation. While the AAE-Recommender was the bestperforming model in the prefetcher evaluation, it is now outperfomed by our two baseline models. The Co-Occurence baseline which was the least performant model in the previous evaluation it now outperforms the other two models. Also notice how the R@2000 from the prefetcher evaluation is not an upper bound for the performance of this baseline. The reason may be that the former recall is averaged sectionwise over all citing papers, while the recall in the prefetcher + reranker evaluation is averaged per citation context over all citing papers. This can lead to a higher recall for the latter by a advantagous distribution of the citation context across sections.

Those results underline the importance of using a strong prefetcher model to achieve high performance on the reranking task. While a global citation recommender has its advantages due to having to be used less often, this has to be exploited using a stronger prefetcher model with higher recall capabilities. While the AAE-Recommender was also choosen due to its high adaptablity considering additional metadata information, its recall performance is not high enough to be competetive to the local citation baseline prefetcher.

6.1.6 Complete Pipeline. As seen in Table 8 the results for the single modules are worse in their individual metrics than on the *Modified S2ORC* dataset. This suggests that our arXiv dataset proves a more difficult dataset for citation recommendation. Some reasons for this would be the very recentness of the papers, to which our pipeline is susceptible due to its id-matching limitations, or remaining LATEXartifacts in the extracted citation contexts. The recall of the cite-worthiness module drops from 68.84% to 49.91% and the precision from 68.26% to 18.44%. This could be due to the different distributions of the labels in the two datasets. While about 26.76% of all citation contexts are cite-worthy in the *Modified S2ORC* dataset, this value drops to 13.61% in our arXiv dataset. Especially the precision seems to be highly susceptible to this fact.

For a better performance in a production environment, the models should therefore be trained on a dataset, which is as close as possible to real-live papers that will be evaluated in the process.

This could explain the gap in the performance difference for the cite-worthiness detection. For the prefetcher & reranker combination another reason may be the insufficient performance of the prefetcher module. When comparing the results of the prefetcher & reranker experiments to the performance in the complete pipeline we can reason that this may be the case.

6.2 Threats to Validity

6.2.1 Datasets. During our experiments we use two datasets namely our arXiv dataset and our *Modified S2ORC* dataset. Because we evaulate a pipeline consisting of multiple modules with different constraints we had to make some limitations regarding our experiments.

arXiv dataset. Due to crawling our dataset from arXiv we rely on the file formats provided by them. Because the LATEXcode is not available for a portion of all papers we can not include them in our experiments. Also for our pipeline SCOPE we had to disregard all papers which do not contain a *.bib* file as their bibliography file. This only became standard in the last few years, before that mostly *.bbl* files are used in the LATEXsource code. Because they do not follow a predictable structure, due to including styling information for the paper, which makes them not trivial to parse, we do not include them in our experiments. As seen in Figure 3, we therefore can mostly evaluate paper after 2018. This can be explained by the fact that the commonly used to tool to preprocess LATEXfor arXiv arxiv_latex_cleaner[12] removes the *.bib* file and only leaves the *.bbl* files. Only recently it got also the option to keep those *.bib* files [13]. This may add a bias in towards more recent papers in our evaluation but does not affect the online usage of pipeline because when writing a paper there will always be a *.bib* file present.

Modified S2ORC *dataset.* While our *Modified S2ORC* dataset follows the citing behaviour found by Bertin et al. [10] (as seen in Table 1), the concrete properties differ in some case from what would be expected when writing a scientific paper. This is the case for the size of the section, which is on average around ten sentences, and would be much higher in a realistic application. This can be explained by the preprocessing steps taken by Wright and Augenstein [75], which removes all entries which cannot be parsed correctly. Also around six years as the average recency of of the cited papers is relatively old for a field like computer science and does not change much between sections. We do not investigate the performance impact of the section distribution in our dataset. Therefore, it is possible that sections with less training instances may produce weaker recommendation results due to an imbalance in the section distribution of the dataset. However, in reality the lengths of different sections also differ. So when sampling from a paper, also the sections will not be represented uniformly. Therefore, our dataset might just reflect this property of a real-live distribution.

6.2.2 Structure Analysis. We had to remove 30% documents from the original data set due to parsing errors or because they did not match. We have adapted our parser to catch cases which occur often. An example of this would be that every word begins with "label":

- "labelintroduction"
- "labelexperiment"
- ...

However, due to the large number of individual problems, this is not possible for the entire data set. By spending more time parsing out the errors, it would be possible to get a more complete data set.

6.2.3 SciBERT Reranker. Our results show that the title is an important part of the input to the reranker. However, there might only be a working title available when performing citation recommendation for a paper that is currently being written. Since our dataset does solely contain already published papers, we did not experiment with working titles but only with final titles. Thus, the creation of a dataset with working titles and the investigation of the effects on the performance of the reranker is a research question for future work.

Due to resource restrictions, we ran the SciBERT Reranker [28] experiments only for a single seed as likewise done by the original authors. As already mentioned above, running the experiments over multiple seeds would give rise to more precise results and reduce the influence of the random seed as an experimental factor.

6.2.4 Complete Pipeline. As stated in Section 4.2.6 our experimental pipeline has some limitations regarding the citation extraction. During the extraction of already cited papers in SCOPE, we rely on them being already present as a candidate paper in our *Modified*

[12] https://github.com/google-research/arxiv-latex-cleaner

[13] https://github.com/google-research/arxiv-latex-cleaner/issues/14

S2ORC dataset. As a consequence, there are papers which can't be mapped to a valid entry in SCOPE and we instead search for a similar paper using BM25.

6.3 Generalization

We have constrained our experiments to the field of computer science. If the pipeline is to be applied to other domains, modifications of the configurations and training might be necessary.

6.3.1 Structure Analysis.

Whether paper structures are repeated in other fields such as biology would have to be verified by experiments. The IMRaD structure, which is similar to our structure, applies to many scientific documents in all fields. We therefore assume that our structure would also work in other domains.

6.3.2 AAE-Recommender. The AAE-Recommender module needs to know all possible candidate papers during training. When applied to another dataset or domain, the module needs to be retrained. Also to perform meaningful prefetching, the module needs at least one already cited paper for its input list.

6.3.3 CiteWorth and SciBERT Reranker. The CiteWorth module and the SciBERT Reranker are both based on Transformer [72] encoder architectures and perform their task solely by means of natural language input. Hence, the models can be utilized with any citing and candidate paper. As such, the SciBERT Reranker is capable of recommending candidate papers that were not part of the training dataset. In general, no re-training of the two modules is required. However, one might consider re-training the modules from time to time with more recent data in order to account for distribution shifts [39].

Another concern is the generalizability of our results achieved in the domain of computer science to other research domains. We refer to Wright and Augenstein [75] and Appendix C.1 for a domain evaluation of the CiteWorth module. Based on these results, we recommend to either train the modules on the respective test domain or alternatively, perform an extensive training over multiple domains in order to transfer to other domains without loss of performance.

6.4 Future Work

6.4.1 Structure Analysis. Some design decisions we made at the beginning of the project turned out to be less than optimal in the end. Although they fulfilled their function sufficiently, better results could have been achieved. These are:

Tree Structure. The tree structure we chose to organize the templates worked for us, but we believe another design could be better suited. Sometimes a list of section headings is matched to multiple templates and no clear allocation can be made. A better approach might be a directed acyclic graph. Trees have one direction (relationships between parents and children) and do not contain cycles. They fit with in the category of Directed Acyclic Graphs. So Trees are DAGs with the restriction that a child can only have one parent.

Natural Language Processing (NLP). In our template we use fixed strings to match to the section sequences, like "Introduction" or "Experiments". This is not optimal, because sometimes the section title consists of a whole sentence, which contains the words we match, for example "detailed experimental setup". A better approach for the future would be to use NLP processing to test the sentences completely. The principle would be that a title is matched as soon as it contains the searched word.

6.4.2 Cite-worthiness Detection. Instead of only predicting the cite-worthiness of individual sentences, one could apply a post-processing to the cite-worthiness detection that accounts for sentences classified as cite-worthy over the whole citing paper. This way, one can prevent recommending duplicate citations. For instance, when a method is named in a paper, the respective reference is commonly only cited at the first and not at every occurrence. Moreover, one may set the threshold θ for classifying a sentence as cite-worthy depending on the section and the amount of citation contexts that should be present in it.

6.4.3 Context-aware Citation Recommendation. We were able to further improve the recommendation performance of Local BM25 by replacing the abstract with the paragraph of the citation context. Local BM25 is regarded as hard-to-beat [28] when employed for prefetching. Hence, it would be interesting to investigate whether proposed alternatives, such as HAtten [28], are still able to outperform this modified input variant of BM25.

For the SciBERT Reranker, we made use of a truncation strategy in order to fit the input to 512 tokens, which is the maximum input length of the SciBERT [7] model. Another strategy would have been to employ a Longformer [8] model instead of the SciBERT model, allowing for a total of $4,096$ input tokens. However, the Longformer model requires more memory resources and is not pre-trained on the scientific domain. When not switching to a multi GPU setup, the former limits the batch size and thereby increases training and evaluation time as in our implementation the actually cited paper needs to be contained in every aggregated batch. Since the SciBERT model is pre-trained on scientific texts, the latter restriction of the Longformer model poses the risk of performance degradation. Due to these constraints, we continued utilizing the SciBERT model in the reranker. Testing the performance of a Longformer model in the task of local citation recommendation is left to future work.

6.4.4 Complete pipeline. The whole pipeline is developed to be used as an online recommendation tool to assists students and researchers with the writing of scientific papers. While we implemented a complete version of the pipeline, we leave the integration of the same into a LaTeX writing tool like Overleaf[14] to future work. Due to the modularity of our pipeline SCOPE, single components can be easily modified and interchanged without much alteration of the existing implementation. Therefore, it is possible to employ other modules that are potentially more suited for the given task.

Furthermore, people writing a scientific document might already know the citation context they are searching a citation for or propose a pool of suitable candidate papers. Hence, the user themselves may replace single modules in the employed pipeline depending on the situation.

Uncertainties due to Sentence-based Citation Contexts. Due to our definition of citation contexts on the sentence-level, it is neither

[14] https://www.overleaf.com

clear how many citations should be placed in the sentence nor where to place them. It is only known whether there is no citation required in the sentence or at least one. Performing a word-wise windowing over the whole document as done by He et al. [29], improves the output in terms of where citations need to be placed. In our pipeline, one could apply this windowing over cite-worthy sentences as a post-hoc step after the CiteWorth module. Still, it is not defined whether one should cite only one or multiple references at a given position. A possible solution is to make the length of the returned citation recommendation list adaptive. By investigating the relevance scores output by the module performing the actual citation recommendation, one could potentially give a recommendation on the amount of citations per identified position. For instance, a certain relevance score threshold or a maximum difference in relevance scores between subsequently ranked candidate papers could be defined and utilized as a cutoff point.

7 CONCLUSION

We explored the different components needed for a complete end-to-end citation recommendation pipeline and provided SCOPE as a modular implementation of the same, employing state-of-the-art components found during our literature research. We analyzed the structure of sections in scientific papers on a large corpus of arXiv papers from the field of computer science and incorporated these results into a synonym dictionary and a template tree. Using those, we evaluated in which ways the section information of scientific papers can be utilized to improve citation recommendation. Lastly, we examined the performance of SCOPE and its individual components and presented those. While we are the first to evaluate a complete end-to-end global context-aware citation recommendation pipeline, our composition leaves room for improvement, especially regarding the usage of a more potent global citation recommender. For reproducability and further research, we provide our code and instructions how to use it on GitHub: https://github.com/Data-Science-2Like/SCOPE

REFERENCES

[1] Zafar Ali, Irfan Ullah, Amin Khan, Asim Ullah Jan, and Khan Muhammad. 2021. An overview and evaluation of citation recommendation models. *Scientometrics* 126, 5 (March 2021), 4083–4119. https://doi.org/10.1007/s11192-021-03909-y

[2] Enago Academy Author. 2021-12. Structure of a Research Paper: Tips to Improve Your Manuscript. https://www.enago.com/academy/tips-effectively-structure-research-paper/

[3] IEEE Author. 2019. Structure Your Paper. https://conferences.ieeeauthorcenter.ieee.org/write-your-paper/structure-your-paper/

[4] TU Wien Academy Author. 2013. How to write a scientific paper. https://www.cg.tuwien.ac.at/resources/HowToWriteAScientificPaper

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:arXiv:1409.0473

[6] Xiaomei Bai, Mengyang Wang, Ivan Lee, Zhuo Yang, Xiangjie Kong, and Feng Xia. 2019. Scientific Paper Recommendation: A Survey. *IEEE Access* 7 (2019), 9324–9339. https://doi.org/10.1109/ACCESS.2018.2890388

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: Pretrained Language Model for Scientific Text. In *EMNLP*. arXiv:arXiv:1903.10676

[8] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[9] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Comput. Linguist.* 22, 1 (mar 1996), 39–71.

[10] Marc Bertin, Iana Atanassova, Yves Gingras, and Vincent Larivière. 2015. The invariant distribution of references in scientific articles. *Journal of the Association for Information Science and Technology* 67, 1 (May 2015), 164–177. https://doi.org/10.1002/asi.23367

[11] Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-Based Citation Recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 238–251. https://doi.org/10.18653/v1/N18-1022

[12] Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. *arXiv preprint arXiv:1802.08301* (2018).

[13] Xiaoyan Cai, Junwei Han, and Libin Yang. 2018. Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[14] Xiaoyan Cai, Yu Zheng, Libin Yang, Tao Dai, and Lantian Guo. 2018. Bibliographic network representation based personalized citation recommendation. *IEEE Access* 7 (2018), 457–467.

[15] Jack Caulfield. 2020-09. How to Structure an Essay. https://www.scribbr.com/academic-essay/essay-structure/

[16] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. 2001. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy*, Peter M. G. Apers, Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Kotagiri Ramamohanarao, and Richard T. Snodgrass (Eds.). Morgan Kaufmann, 109–118. http://www.vldb.org/conf/2001/P109.pdf

[17] Tao Dai, Li Zhu, Yifan Wang, Hongfei Zhang, Xiaoyan Cai, and Yu Zheng. 2018. Joint model feature regression and topic learning for global citation recommendation. *IEEE Access* 7 (2018), 1706–1720.

[18] 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[19] Daniel Duma and Ewan Klein. 2014. Citation Resolution: A method for evaluating context-based citation recommendation systems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 358–363. https://doi.org/10.3115/v1/P14-2059

[20] Travis Ebesu and Yi Fang. 2017. Neural Citation Network for Context-Aware Citation Recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1093–1096. https://doi.org/10.1145/3077136.3080730

[21] Steffen Epp, Marcel Hoffmann, Nicolas Lell, Michael Mohr, and Ansgar Scherp. 2021. A Machine Learning Pipeline for Automatic Extraction of Statistic Reports and Experimental Conditions from Scientific Papers. arXiv:2103.14124 [cs.DL]

[22] Michael Färber and Adam Jatowt. 2020. Citation recommendation: approaches and datasets. *International Journal on Digital Libraries* 21, 4 (Aug. 2020), 375–405. https://doi.org/10.1007/s00799-020-00288-2

[23] Michael Färber, Alexander Thiemann, and Adam Jatowt. 2018. CITEWERTs: A System Combining Cite-Worthiness with Citation Recommendation. In *Lecture Notes in Computer Science*. Springer International Publishing, 815–819. https://doi.org/10.1007/978-3-319-76941-7_82

[24] Michael Färber, Alexander Thiemann, and Adam Jatowt. 2018. To Cite, or Not to Cite? Detecting Citation Contexts in Text. In *Lecture Notes in Computer Science*. Springer International Publishing, 598–603. https://doi.org/10.1007/978-3-319-76941-7_50

[25] Lorenz Froihofer. 2013. *How to write a computer science paper.* https://www.froihofer.net/en/students/how-to-write-a-computer-science-paper.html

[26] Lukas Galke, Ahmed Saleh, and Ansgar Scherp. 2017. Word embeddings for practical information retrieval. *INFORMATIK 2017* (2017).

[27] E. Mark Gold. 1967. Language Identification in the Limit. *Inf. Control.* 10, 5 (1967), 447–474. https://doi.org/10.1016/S0019-9958(67)91165-5

[28] Nianlong Gu, Yingqiang Gao, and Richard H. R. Hahnloser. 2022. Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-Based Reranking. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022 (Lecture Notes in Computer Science, Vol. 13185)*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer, 274–288. https://doi.org/10.1007/978-3-030-99736-6_19

[29] Qi He, Daniel Kifer, Jian Pei, Prasenjit Mitra, and C. Lee Giles. 2011. Citation Recommendation without Author Supervision. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining* (Hong Kong, China) *(WSDM '11)*. Association for Computing Machinery, New York, NY, USA, 755–764. https://doi.org/10.1145/1935826.1935926

[30] Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. 2010. Context-Aware Citation Recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. Association for Computing Machinery, New York, NY, USA, 421–430. https://doi.org/10.1145/

1772690.1772734

[31] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications* 13, 4 (1998), 18–28. https://doi.org/10.1109/5254.708428

[32] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (11 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf

[33] Wenyi Huang, Zhaohui Wu, Prasenjit Mitra, and C. Lee Giles. 2014. RefSeer: A citation recommendation system. In *IEEE/ACM Joint Conference on Digital Libraries*. IEEE, 371–374. https://doi.org/10.1109/jcdl.2014.6970192

[34] Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with BERT and graph convolutional networks. *Scientometrics* 124, 3 (July 2020), 1907–1922. https://doi.org/10.1007/s11192-020-03561-y

[35] Nayef Jomaa Jomaa and Siti Jamilah Bidin. 2017. Perspectives of EFL Doctoral Students on Challenges of Citations in Academic Writing. *Malaysian Journal of Learning and Instruction (MJLI)* 14, 2 (2017), 177–209. https://eric.ed.gov/?id=EJ1166743

[36] David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2016. Citation Classification for Behavioral Analysis of a Scientific Field. https://doi.org/10.48550/ARXIV.1609.00435

[37] Gail Kaiser, Craig Partridge, Sumit Roy, Eric Siegel, Sal Stolfo, Luca Trevisan, Yechiam Yemini, Erez Zadok, and João Craveiro. 2021. *Writing Technical Articles*. https://www.cs.columbia.edu/~hgs/etc/writing-style.html

[38] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[39] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 5637–5664. https://proceedings.mlr.press/v139/koh21a.html

[40] Kevin B. Korb. 1999. *Research Writing in Computer Science*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.148&rep=rep1&type=pdf

[41] Christin Katharina Kreutz and Ralf Schenkel. 2022. Scientific Paper Recommendation Systems: a Literature Review of recent Publications. arXiv:arXiv:2201.00682

[42] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2267–2273.

[43] R. B. Lamptey and H. Atta-Obeng. 2012. Challenges with Reference Citations among Postgraduate Students at the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana. *Journal of Science and Technology* 32, 3 (2012), 69–80. https://doi.org/10.4314/just.v32i3.8

[44] Avishay Livne, Vivek Gokuladas, Jaime Teevan, Susan T. Dumais, and Eytan Adar. 2014. CiteSight: Supporting Contextual Citation Recommendation Using Differential Search. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 807–816. https://doi.org/10.1145/2600428.2609585

[45] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4969–4983. https://doi.org/10.18653/v1/2020.acl-main.447

[46] Jessica Lubzyk, Christiane Fitzke, Sabine Frey, Dirk Funck, Hans-Karl Hauffe, Sylvia Lepp, Dana Loewy, Rüdiger Reinhardt, Uwe Rothfuß, Kerstin Schramm, and Carola Pekrun. 2018-07. A Guide to Writing an Academic Paper. https://www.hfwu.de/fileadmin/user_upload/IBIS/Leitfaeden/EN_Guide_to_Writing_an_Academic_Paper.pdf

[47] Chris A. Mack. 2014. *How to Write a Good Scientific Paper: Structure and Organization*. https://www.spiedigitallibrary.org/journals/journal-of-micro-nanopatterning-materials-and-metrology/volume-13/issue-04/040101/How-to-Write-a-Good-Scientific-Paper--Structure-and/10.1117/1.JMM.13.4.040101.full?SSO=1

[48] Chris A. Mack. 2018. *How to Write a Good Scientific Paper*. https://spie.org/samples/9781510619142.pdf

[49] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).

[50] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2009. *An introduction to information retrieval* (online ed.). Cambridge University Press. http://www.informationretrieval.org

[51] Zoran Medić and Jan Snajder. 2020. Improved Local Citation Recommendation Based on Context Enhanced with Global Information. In *Proceedings of the First Workshop on Scholarly Document Processing*. Association for Computational Linguistics, Online, 97–103. https://doi.org/10.18653/v1/2020.sdp-1.11

[52] Zoran Medic and Jan Snajder. 2020. A Survey of Citation Recommendation Tasks and Methods. *J. Comput. Inf. Technol.* 28, 3 (2020), 183–205. https://doi.org/10.20532/cit.2020.1005160

[53] Ion Muslea, Steven Minton, and Craig A. Knoblock. 1999. A Hierarchical Approach to Wrapper Induction. In *Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS 1999, Seattle, WA, USA, May 1-5, 1999*, Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw (Eds.). ACM, 190–197. https://doi.org/10.1145/301136.301191

[54] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[55] Rodrigo Nogueira, Zhiying Jiang, Kyunghyun Cho, and Jimmy Lin. 2020. Navigation-based candidate expansion and pretrained language models for citation recommendation. *Scientometrics* 125, 3 (2020), 3001–3016.

[56] Charlene Polio and Ling Shi. 2012. Perceptions and beliefs about textual appropriation and source use in second language writing. *Journal of Second Language Writing* 21, 2 (2012), 95–101. https://doi.org/10.1016/j.jslw.2012.03.001

[57] Quora. 2018. *What Is The Best Way To Write A Computer Science Research Paper?* https://www.forbes.com/sites/quora/2018/01/08/what-is-the-best-way-to-write-a-computer-science-research-paper/?sh=11f1c3bc5265

[58] Sidney Redner. 1998. How popular is your paper? An empirical study of the citation distribution. *The European Physical Journal B-Condensed Matter and Complex Systems* 4, 2 (1998), 131–134.

[59] Francesco Ricci. 2014/2015. *Research Methods and Paper Writing*. https://www.inf.unibz.it/~calvanese/teaching/2014-11-PhD-RM/RM-2014-M3-ricci.pdf

[60] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (apr 2009), 333–389. https://doi.org/10.1561/1500000019

[61] Agata Rotondi, Angelo Di Iorio, and Freddy Limpens. 2018. Identifying Citation Contexts: a Review of Strategies and Goals. In *Fifth Italian Conference on Computational Linguistics CLiC-it*. Academai University Press, 335–341. https://doi.org/10.4000/books.aaccademia.3594

[62] Henry Small. 1973. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for information Science* 24, 4 (1973), 265–269.

[63] Luciana B Sollaci and Mauricio G Pereira. 2004. The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. *Journal of the Medical Library Association* 92, 3 (July 2004), 364–367.

[64] Anthony G Stacey. 2020. Robust parameterisation of ages of references in published research. *Journal of Informetrics* 14, 3 (2020), 101048. https://doi.org/10.1016/j.joi.2020.101048

[65] Kazunari Sugiyama, Tarun Kumar, Min-Yen Kan, and Ramesh C. Tripathi. 2010. Identifying citing sentences in research papers using supervised learning. In *2010 International Conference on Information Retrieval Knowledge Management (CAMP)*. 67–72. https://doi.org/10.1109/INFRKM.2010.5466945

[66] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? arXiv:arXiv:1905.05583

[67] Iman Tahamtan and Lutz Bornmann. 2018. Core elements in the process of citing publications: Conceptual overview of the literature. *Journal of Informetrics* 12, 1 (2018), 203–216. https://doi.org/10.1016/j.joi.2018.01.002

[68] Xuewei Tang, Xiaojun Wan, and Xun Zhang. 2014. Cross-Language Context-Aware Citation Recommendation in Scientific Articles. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 817–826. https://doi.org/10.1145/2600428.2609564

[69] Patricia M. Hudelson Thomas V. Perneger. 2004. Writing a research article: advice to beginners. https://academic.oup.com/intqhc/article/16/3/191/1814554

[70] Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to BM25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*. 58–65.

[71] Iacopo Vagliano, Lukas Galke, and Ansgar Scherp. 2021. Recommendations for Item Set Completion: On the Semantics of Item Co-Occurrence With Data Sparsity, Input Size, and Input Modalities. arXiv:2105.04376 [cs.IR]

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[73] Xiaoguang Wang, Ningyuan Song, Huimin Zhou, and Hanghang Cheng. 2021. The representation of argumentation in scientific papers: A comparative analysis of two research areas. *Journal of the Association for Information Science and Technology* (Oct. 2021). https://doi.org/10.1002/asi.24590

[74] Jennifer Widom. 2006-01. Tips for Writing Technical Papers. https://cs.stanford.edu/people/widom/paper-writing.html. Accessed: 2022-07-22.

[75] Dustin Wright and Isabelle Augenstein. 2021. CiteWorth: Cite-Worthiness Detection for Improved Scientific Document Understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 1796–1807. https://doi.org/10.18653/v1/2021.findings-acl.157

[76] Libin Yang, Zeqing Zhang, Xiaoyan Cai, and Lantian Guo. 2019. Citation recommendation as edge prediction in heterogeneous bibliographic network: a network representation approach. *IEEE Access* 7 (2019), 23232–23239.

[77] Yang Zhang and Qiang Ma. 2020. Citation recommendations considering content and structural context embedding. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 1–7.

[78] Junbo Zhao, Michael Mathieu, and Yann LeCun. 2016. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126* (2016).

A STRUCTURE ANALYSIS APPENDIX

A.1 Examples of sequence structures not processed by us

- [’introduction and relevance’, ’ultrasound scanning in a nutshell’, ’digital ultrasound beamforming’, ’deep learning opportunity’, ’deep learning architecture for ultrasound beamforming’, ’training strategy and data’, ’new research opportunity’]
- [’introduction’, ’natural language semantic parsing’, ’program synthesis code generation’, ’evolution of semantic parsing’, ’supervision in semantic parsing’, ’modern advance in neural semantic parsing for code generation’, ’future direction’, ’broader impact’]
- [’introduction’, ’preliminary’, ’bridging the gap a unified view’, ’experiment’, ’discussion’, ’computation of tunable parameter’, ’full result on different bottleneck dimension’]
- [’introduction’, ’method’, ’experiment’, ’discussion’, ’detailed experimental setup’, ’other variant’, ’additional result’]
- [’introduction’, ’related work’, ’preliminary’, ’gender stereotype in word embeddings’, ’geometry of gender and bias’, ’debiasing algorithm’, ’determining gender neutral word’, ’debiasing result’, ’other type of bias’, ’discussion’, ’generating analogy’, ’learning the linear transform’, ’detail of gender specific word base set’, ’questionnaire for generating gender stereotypical word’, ’questionnaire for generating gender stereotypical analogy’, ’questionnaire for rating stereotypical analogy’, ’questionnaire for ranking stereotypical analogy’, ’analogy generated by word embeddings’]
- [’datasets and performance’, ’discussion and open problem’, ’related work’, ’method’, ’introduction’]
- [’introduction’, ’formatting your paper’, ’page title section’, ’typestyle and font’, ’major heading’, ’printing your paper’, ’page numbering’, ’illustration graph and photograph’, ’footnote’, ’copyright form’, ’related work’, ’reference’]
- [’introduction’, ’preliminary’, ’prediction of pack’, ’algorithm for prediction of pack’, ’a mix loss lower bound’, ’experiment’]
- [’related work’, ’kinn the proposed framework’, ’experiment’, ’conclusion’]
- [’introduction’, ’online model selection’, ’discussion and further direction’, ’proof’]
- [’introduction’, ’author name’, ’affiliation’, ’mapping author to affiliation’, ’email’]

A.2 Synonym Dictionary

```
 ###################################
# Mappings we defined at the start of the experiment
 ###################################
 "introduction": "introduction",
 "introduction": "intro",
 "introduction": "overview",
 "introduction": "motivation",
 "introduction": "problem motivation",
 "related work": "related work",
 "related work": "previous work",
 "related work": "literature",
 "related work": "background",
 "related work": "literature review",
 "related work": "state of the art",
 "related work": "current state of research",
 "related work": "requirement",
 "experiment": "experiment",
 "experiment": "experimental result",
 "experiment": "experimental setup",
 "experiment": "result",
 "experiment": "result and analysis",
 "experiment": "evaluation",
 "experiment": "performance evaluation",
 "experiment": "experiment and result",
 "experiment": "analysis",
 "method": "method",
 "method": "methodology",
 "method": "material and method",
 "method": "proposed method",
 "method": "evaluation methodology",
 "method": "procedure",
 "method": "implementation",
 "method": "experimental design",
 "method": "implementation detail",
 "method": "system model",
 "discussion": "discussion",
 "discussion": "limitation",
 "discussion": "result and discussion",
 "conclusion": "conclusion",
 "conclusion": "future work",
 "conclusion": "summary",
 "conclusion": "discussion and conclusion",
 "conclusion": "conclusion and outlook",
 "conclusion": "conclusion and future work",
 "conclusion": "concluding remark",
 ###################################
 # Mappings learned through training
 ###################################
 "result": "empirical result",
 "experiment": "experiment result",
"conclusion": "conclusion limitation and future work",
 "conclusion": "conclusion and future direction",
 "related work": "relation to prior work",
 "related work": "background and related work",
 "experiment": "experimental evaluation",
```

Table 9: Citation recommendation performance on the validation split of the *ACL-200_Reranker* and *S2ORC_Reranker* dataset, respectively, with the oracle variant of Local BM25 as the prefetcher and $k = 2,000$. The sentence-based citation context was represented in the *default* variant. For all metrics, we report the mean and standard error (value in brackets) over the $9,381$ and $8,119$ citation contexts respectively.

	ACL-200_Reranker		**ModifiedS2ORC_Reranker**	
Epoch	MRR	R@10	MRR	R@10
1	$0.498_{(0.004)}$	$0.749_{(0.004)}$	$0.580_{(0.005)}$	$0.774_{(0.004)}$
2	$0.548_{(0.004)}$	$0.796_{(0.004)}$	$0.588_{(0.005)}$	$0.785_{(0.004)}$
3	$0.563_{(0.004)}$	$0.806_{(0.004)}$	$0.600_{(0.005)}$	$0.797_{(0.004)}$
4	$0.570_{(0.004)}$	$0.808_{(0.004)}$	$0.607_{(0.005)}$	$0.799_{(0.004)}$
5	$0.578_{(0.004)}$	$0.812_{(0.004)}$	$0.613_{(0.005)}$	$0.803_{(0.004)}$

```
"related work": "technical background",
"method": "model",
"experiment": "result and evaluation",
"discussion": "discussion and future work",
"related work": "related work and background",
"conclusion": "future work and conclusion",
"conclusion": "conclusion and discussion",
"introduction": "introduction and motivation",
"experiment": "evaluation and result",
"method": "proposed approach",
"conclusion": "related work and conclusion",
"conclusion": "conclusion and limitation",
"related work": "motivation and related work",
"introduction": "introduction and related work",
"conclusion": "summary and conclusion",
"related work": "related literature",
"conclusion": "conclusion and perspective",
"introduction": "introduction and background",
"method": "proposed methodology",
"related work": "review of previous method",
"discussion": "discussion and outlook"
```

B EXPERIMENTAL RESULTS FOR OPTIMIZING THE HYPERPARAMETERS OF THE SCIBERT RERANKER

For the SciBERT Reranker [28] all hyperparameters except the amount of training epochs were given by the authors. Hence, we searched over {1, 2, 3, 4, 5} as the number of epochs for the *ModifiedS2ORC_Reranker* as well as the *ACL-200_Reranker* dataset. In both cases, we applied the *default* input variant to the SciBERT model. The MRR and R@10 on the respective validation split are reported after every training epoch in Table 9. The highest MRR and R@10 were achieved for both datasets after the fifth epoch.

Table 10: Cite-worthiness detection performance on the test split of the CiteWorth [75] dataset. The longformer-ctx models were trained on the given domains and tested in the computer science domain. The binary recall (R), precision (P), and F1 metric for the cite-worthy class are given in percent.

training domains	R	P	F1
Computer Science	71.73	55.23	62.41
all	71.20	57.59	63.68
all except Computer Science	71.13	56.14	62.75

C ADDITIONAL EXPERIMENTS

C.1 Domain Evaluation of the Cite-worthiness Detection

The CiteWorth [75] dataset entails paragraphs sampled equally from ten different research domains. The training split contains a total of $945,426$ sentences. We focused our work on the computer science domain, i. e., we aim at good test results in this domain. Wright and Augenstein [75] investigated to what extend a cite-worthiness detection trained on one of the domains can generalize to another domain. As expected, they achieved the best results when training the model with papers from the test domain. Additionally, we investigated whether training on all domains of the CiteWorth dataset can improve the performance in the computer science domain. As an ablation study, we also trained the model on all domains except computer science. Our results for training the *ctx* variant on all domains, computer science, and all domains except computer science are summarized in Table 10. For testing on the computer science domain only, the highest F1 and precision was achieved when training on all domains. The best recall resulted from training on the computer science domain only, although the difference in recall is smaller than the one in precision and F1. Since training time is increased by factor 10 when training on all domains compared to training on the computer science domain only, we concluded that the additional training effort is not worth the improvement in performance. Moreover, as discussed in Section 6.1, a high recall is slightly more important to us than a high precision since the user is part of the loop and needs to accept or reject recommendations of the system.

C.2 AAE-Recommender Performance on different train-test split

Because the AAE-Recommender can only use papers of the training set as candidate papers the idea arises that the AAE-Recommender performs better when a the size of the training data increases by using a different year to split between training and test data. We test this hypothesis for notable performance differences by using a different train-test split. The corresponding results are in Table 11. As it can be seen the performance does not increase but rather degrades.

Table 11: Performance Comparison Old vs New Split. Old refers to using all paper until 2018 as training data, while the new split also uses the year 2019 for training data to allow for more candidate papers. Datathreshold was 1. For both metrics we report the mean and standard deviation over 4,199 (old) and 260 (new) test records.

	$d = 0.2$		$d = 0.5$		$d = 0.8$	
	MRR	R@2000	MRR	R@2000	MRR	R@2000
AAE-title (new)	$0.006_{(0.037)}$	$0.315_{(0.450)}$	$0.024_{(0.105)}$	$0.335_{(0.434)}$	$0.038_{(0.139)}$	$0.329_{(0.425)}$
AAE-section-title (new)	$0.006_{(0.038)}$	$0.282_{(0.431)}$	$0.025_{(0.105)}$	$0.315_{(0.432)}$	$0.039_{(0.141)}$	$0.320_{(0.424)}$
AAE-none (old)	$0.012_{(0.090)}$	$0.343_{(0.092)}$	$0.029_{(0.133)}$	$0.335_{(0.435)}$	$0.045_{(0.166)}$	$0.327_{(0.420)}$
AAE-title (old)	$0.012_{(0.092)}$	$0.360_{(0.460)}$	$0.028_{(0.135)}$	$0.350_{(0.440)}$	$0.043_{(0.165)}$	$0.340_{(0.424)}$
AAE-section-title (old)	$0.012_{(0.087)}$	$0.377_{(0.466)}$	$0.028_{(0.131)}$	$0.360_{(0.443)}$	$0.044_{(0.162)}$	$0.348_{(0.427)}$
AAE-section (old)	$0.012_{(0.08)}$	$0.364_{(0.462)}$	$0.029_{(0.129)}$	$0.349_{(0.440)}$	$0.045_{(0.159)}$	$0.343_{(0.425)}$

C.3 Comparison of the AAE-Recommender on popular global citation recommendation dataset

A comparison of the AAE-Recommender with different state-of-the-art global citation recommender can be seen in Table 12. Note that for the other modules the values are taken from their respective paper and apply different kind of preprocessing like removing multiple categories of the dataset. The results should therefore be taken with a grain of salt.

D CITE-WORTHINESS DETECTION EXPERIMENTS ON A PREVIOUS VERSION OF OUR MODIFIED S2ORC DATASET

We have conducted the cite-worthiness detection experiments with the CiteWorth module on a previous version of the *Modified S2ORC* dataset. This version entailed more citing papers from the field of computer science and was a valid data basis for performing cite-worthiness detection, though it was not suited as a common data basis for all modules in the pipeline. We performed the same preprocessing steps and experimental procedure including hyperparameter selection as given in Section 4. In order to distinguish the *ModifiedS2ORC_CiteWorth* dataset from this version, we term the latter *prevModifiedS2ORC_CiteWorth*. The *prevModifiedS2ORC_CiteWorth* dataset contains nearly twice as many paragraphs for training and around 50% more paragraphs in the test splits than the *ModifiedS2ORC_CiteWorth* dataset.

D.1 Results

Table 13 lists the results of the cite-worthiness detection experiments with the CiteWorth module on the *prevModifiedS2ORC_CiteWorth* dataset. As before on the *ModifiedS2ORC_CiteWorth* dataset (see Section 5.2), the results for the *ctx* variant on the *test_conform* data are slightly worse than in the original work by Wright and Augenstein [75], the Longformer models outperform the baselines irrespective of the input variant and the selected test data, and the F1 metric shows the same trend over the different kinds of test data.

In contrast, the results differ when comparing the *ctx* variant to the three *ctx-section* variants: With the *ModifiedS2ORC_CiteWorth* dataset, adding the section type to the input leads to an improvement in precision and a decrease in recall performance. With the *prevModifiedS2ORC_CiteWorth* dataset, however, the recall improves between 0.08% and 0.90% compared to the *ctx* variant, while the effect on precision depends on the specific *ctx-section* and test data variant. Adding the section type to the input consistently improves the performance of the classifiers in terms of F1, which is not the case with the *ModifiedS2ORC_CiteWorth* dataset. However, the improvement is rather minor with less than 0.4% for the *test_all* dataset. The class-weighted baseline was able to improve up to 0.89% in terms of F1 by adding the section information as a prior.

D.2 Discussion

In Section 6.1.2, we reasoned that the Longformer model should in theory be able to improve more than the baseline when adding section information. Our experiments with the *ModifiedS2ORC_CiteWorth* dataset only support this theory in terms of precision. The results with the *prevModifiedS2ORC_CiteWorth* dataset, however, do not support this theory. Based on the results with the *prevModifiedS2ORC_CiteWorth* dataset, we hypothesise that a complete paragraph contains enough information in order to infer the respective section type. Hence, it cannot be excluded that the Longformer model implicitly utilizes section types in the *ctx* variant. As a first attempt of validating our assumption, we performed the same experiments with the Longformer model on the sentence-level, i. e., we input single sentences instead of complete paragraphs. We term these variants *solo* instead of *ctx*. It is still possible that even a single sentence is sufficient in order to implicitly infer section information, albeit the task is more challenging due to the limited input. In our experiments with the *prevModifiedS2ORC_CiteWorth* dataset, the *solo* models can improve more by adding section information than the *ctx* ones. For the *test_all* data, for instance, the F1 value increases more than 0.5%. More details on the experimental setup and the results of the *solo* variants are given below.

D.3 Solo Variants

The *solo* variants of the CiteWorth module utilize single sentences as an input instead of complete paragraphs. As a consequence the *[CLS]* token is forwarded from the Longformer [8] model to the subsequent classification network, instead of the *[SEP]* tokens. Transferring the former paragraph-level input variants to this input mode, results in the following input variants, where “sec” and “sent” are short for section type and sentence respectively:

Table 12: Performance of Global Citation Recommender on popular Datasets.

	AAN		ACM		DBLP	
	MRR	R@100	MRR	R@100	MRR	R@100
GAN-HBNR [13]	0.312	0.787	-	-	0.289	0.758
BNR [14]	0.296	0.762	-	-	0.295	0.729
Doc2Vec [77]	0.265	0.741	-	-	0.263	0.703
NREP [76]	-	0.775	-	-	-	0.736
Ranking SciBERT [55]	-	-	-	-	0.714	0.892
AAE with title	0.122	0.413	0.140	-	0.140	-

Table 13: Cite-worthiness detection performance on the three test splits of the *prevModifiedS2ORC_CiteWorth* dataset. For all metrics, we report the mean and standard deviation (value in brackets) over five different seeds in percent.

	test_conform			test_non-conform			test_all		
	R	P	F1	R	P	F1	R	P	F1
ctx	72.59$_{(1.12)}$	58.08$_{(1.03)}$	64.52$_{(0.23)}$	68.14$_{(1.01)}$	62.57$_{(1.41)}$	65.22$_{(0.41)}$	67.88$_{(1.17)}$	66.84$_{(1.17)}$	67.34$_{(0.21)}$
ctx-section-extra	72.67$_{(0.89)}$	**58.24$_{(0.97)}$**	64.65$_{(0.31)}$	68.75$_{(1.20)}$	**63.32$_{(1.08)}$**	**65.91$_{(0.27)}$**	68.13$_{(1.13)}$	**67.18$_{(0.87)}$**	67.64$_{(0.28)}$
ctx-section-always	73.16$_{(0.72)}$	57.93$_{(0.45)}$	**64.66$_{(0.09)}$**	**68.84$_{(0.61)}$**	62.58$_{(0.75)}$	65.55$_{(0.32)}$	**68.78$_{(0.62)}$**	66.72$_{(0.53)}$	**67.73$_{(0.19)}$**
ctx-section-first	**73.27$_{(0.78)}$**	57.86$_{(0.56)}$	64.65$_{(0.14)}$	68.62$_{(0.94)}$	62.68$_{(0.92)}$	65.51$_{(0.27)}$	68.36$_{(0.87)}$	66.73$_{(0.77)}$	67.53$_{(0.18)}$
chance baseline[a]	50.21$_{(0.68)}$	28.06$_{(0.36)}$	36.00$_{(0.47)}$	49.72$_{(0.44)}$	29.40$_{(0.12)}$	36.95$_{(0.21)}$	49.89$_{(0.32)}$	34.17$_{(0.21)}$	40.56$_{(0.25)}$
class-weighted baseline[b]	26.16$_{(0.39)}$	27.93$_{(0.52)}$	27.01$_{(0.45)}$	25.92$_{(0.40)}$	29.32$_{(0.24)}$	27.52$_{(0.32)}$	25.98$_{(0.40)}$	34.07$_{(0.37)}$	29.48$_{(0.39)}$
class-weighted baseline + section[c]	26.41$_{(0.40)}$	28.58$_{(0.43)}$	27.45$_{(0.41)}$	26.38$_{(0.46)}$	30.16$_{(0.35)}$	28.14$_{(0.39)}$	26.79$_{(0.39)}$	35.06$_{(0.37)}$	30.37$_{(0.37)}$

Values of ϕ determined by the baselines. ϕ of the times, a sentence is classified as cite-worthy.

[a] $\phi = 50\%$, [b] $\phi \approx 26.11\%$, [c] $\phi(\text{introduction}) \approx 27.22$, $\phi(\text{related work}) \approx 30.03\%$, $\phi(\text{method}) \approx 19.76\%$, $\phi(\text{experiment}) \approx 17.00\%$, $\phi(\text{discussion}) \approx 20.27\%$, $\phi(\text{conclusion}) \approx 20.84\%$

- *solo*[15]: Does not contain any information about the section, i. e., *[CLS]* sent *[SEP]*
- *solo-section-extra*: The section type is added in front of the input sequence followed by a *[SEP]* token, i. e., *[CLS]* sec *[SEP]* sent *[SEP]*
- *solo-section-firstalways*: The section type is added in front of the sentence, i. e., *[CLS]* sec sent *[SEP]*.

Just as for the *ctx* variants, we reuse the hyperparameters of the *Longformer-solo* model optimized by Wright and Augenstein [75] on their CiteWorth dataset:

- batch size: 4
- number of epochs: 3
- learning rate: 0.000001351
- triangular learning rate warmup steps: 300
- weight decay: 0.1
- dropout probability: 0.1

The results of the *solo* variant experiments are shown in Table 14. Adding the section type to the input leads to improvements in precision, recall, and F1 over all three test datasets. Among the three *solo* variants, the *solo-section-firstalways* variant overall performs the best. Though, any of the *ctx* variants (see Table 13 for those results) utilizing a complete paragraph for the sentence-wise cite-worthiness classification performs better than the *solo* variants, which was also found by Wright and Augenstein [75] without adding the section type.

[15] The *solo* variant is adopted from the original work by Wright and Augenstein [75].

Table 14: Cite-worthiness detection performance on the three test splits of the *prevModifiedS2ORC_CiteWorth* dataset. Instead of utilizing the whole paragraph as an input to the Longformer model (see Table 13 for those results), we only input single sentences. For all metrics, we report the mean and standard deviation (value in brackets) over five different seeds in percent.

	test_conform			**test_non-conform**			**test_all**		
	R	P	F1	R	P	F1	R	P	F1
solo	62.32$_{(1.16)}$	56.26$_{(0.88)}$	59.12$_{(0.16)}$	58.12$_{(0.82)}$	61.89$_{(0.99)}$	59.93$_{(0.37)}$	62.48$_{(1.04)}$	65.28$_{(0.85)}$	63.84$_{(0.25)}$
solo-section-extra	62.64$_{(1.90)}$	**56.44**$_{\mathbf{(1.08)}}$	**59.35**$_{\mathbf{(0.23)}}$	59.41$_{(1.17)}$	62.28$_{(1.22)}$	60.79$_{(0.45)}$	63.36$_{(1.47)}$	65.51$_{(1.07)}$	64.40$_{(0.33)}$
solo-section-firstalways	**62.67**$_{\mathbf{(1.89)}}$	56.37$_{(1.16)}$	59.32$_{(0.18)}$	**59.45**$_{\mathbf{(1.18)}}$	**62.46**$_{\mathbf{(1.26)}}$	**60.90**$_{\mathbf{(0.61)}}$	**63.42**$_{\mathbf{(1.52)}}$	**65.56**$_{\mathbf{(1.13)}}$	**64.45**$_{\mathbf{(0.38)}}$