Improved Local Citation Recommendation Based on Context Enhanced with Global Information

Zoran Medić and **Jan Šnajder**
Text Analysis and Knowledge Engineering Lab
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
{zoran.medic,jan.snajder}@fer.hr

Abstract

Local citation recommendation aims at finding articles relevant for given citation context. While most previous approaches represent context using solely text surrounding the citation, we propose enhancing context representation with global information. Specifically, we include citing article's title and abstract into context representation. We evaluate our model on datasets with different citation context sizes and demonstrate improvements with globally-enhanced context representations when citation contexts are smaller.

1 Introduction

The number of published scientific articles has been growing rapidly: it surpassed 50 million in 2009 (Jinha, 2010) and the global publication rate is still growing (Ware and Mabe, 2015). As a consequence, scholars are finding it increasingly difficult to keep up with relevant research. The problem can be alleviated by citation recommendation (CR) systems, which help researchers writing articles find published work that they might consider citing.

Approaches to CR come in two types: *global* (Bethard and Jurafsky, 2010; Ren et al., 2014; Guo et al., 2017; Bhagavatula et al., 2018) and *local* (He et al., 2010; Huang et al., 2012; Ebesu and Fang, 2017; Dai et al., 2019). Global CR models typically take the article's abstract or the entire text as input (the *citing* article) and output a list of relevant articles (the candidates for *cited* articles). In contrast, local CR models take an excerpt of the citing article (the citation context) as input and recommend articles that may be cited specifically in that context.[1] Although the cited article is already described by the citation context which the local CR models use to find relevant articles, we hypothesize that these models could still benefit from global information on the citing article, which provides a broader context for specific citation. For instance, consider the following citation context:[2]

> *Other approaches include for example clustering-based algorithms, such as the one presented in* CITATION*, or techniques which rely on building a statistical language model to rank KPs, like the one presented in* CITATION*.*

Looking at the context alone, it is difficult to tell which articles were cited at CITATIONs. However, with global information (citing article's text) available, the model could narrow the search.

In this work, we address the task of local CR and experiment with enhancing the citation context with citing article's global information, more specifically with its title and abstract. We propose a model that produces the final recommendation score as a combination of semantic (based on the match between citing article's citation context and cited article's content) and bibliographic relevance (based on articles' popularity in the scientific community). Our evaluation on two datasets with different citation context sizes shows that enhancing citation context with global information helps when the citation context is smaller. We also show that inclusion of bibliographic relevance leads to better results compared to model with semantic relevance only. The contribution of our work is twofold: (1) we present a model that includes global information into local CR and obtains competitive results compared to previous work (Ebesu and Fang, 2017) and (2) we show that inclusion of global information besides citation context helps when citation contexts are smaller.

[1] Most work on local CR uses contexts from published articles with citations masked out. While this is a somewhat artificial setup as those articles are biased toward existing citations, local CR can nonetheless be used in this setup to recommend additional citations.

[2] Excerpt is from *Evaluating anaphora and coreference resolution to improve automatic keyphrase extraction* of Basaldella et al. (2016). The articles cited are *Clustering to find exemplar terms for keyphrase extraction* of Liu et al. (2009) and *A language model approach to keyphrase extraction* of Tomokiyo and Hurst (2003).

Proceedings of the First Workshop on Scholarly Document Processing, pages 97–103

https://doi.org/10.18653/v1/P17

2 Related Work

Compared to global CR, local CR has attracted less attention, presumably as there is fewer datasets with extracted citation contexts. A comprehensive overview of local CR is (Färber and Jatowt, 2020).

The task of local CR was introduced by He et al. (2010), who used TF-IDF representations of contexts and cited articles in a vector similarity based setup. He et al. (2011) proposed a model that detects contexts in an article's text and recommends citations using TF-IDF based matching with other contexts extracted from a corpus of articles. Huang et al. (2012) framed the task as statistical machine translation from context to cited article. Livne et al. (2014) built a system that provides recommendations while manuscript is being written. The system is based on a number of hand-crafted features extracted from both citation context and the rest of article's content.

Recently, deep learning models were proposed for the task. Huang et al. (2015) presented a neural probabilistic model that embeds words from the context and all the articles from the corpus into a shared embedding space. A neural module uses the so-obtained embeddings for determining the probability of citing an article in a given context. The model of Ebesu and Fang (2017) encodes citation context and decodes it into the title of cited article, using also the information about the authors of citing and cited articles. Dai et al. (2019) use stacked denoising autoencoders for representing cited articles and bidirectional LSTMs for citation context's embedding. Our approach extends on these models by enhancing the representation of citation context with global information. While there has been work that proposed adding global information to context representation (He et al., 2010; Livne et al., 2014), to the best of our knowledge, no such deep learning model has been proposed.

3 Model

The proposed model takes five inputs: (1) citation context text, (2) citing article's title and abstract (henceforth: citing article's text), (3) cited article title and abstract (henceforth: candidate article's text), (4) a list of cited article authors, and (5) the number of cited article citations per last y years, together with the total number of citations. The output of the model is a recommendation score indicating whether candidate article should be cited in input context.

Two modules make up the model: the semantic module and the bibliographic module. We depict the two modules in Figures 1 and 2. The final recommendation score is a weighted sum of the scores produced by the two modules. The intuition behind the weighted sum of the scores is that, depending on the context, the authors might prefer to cite articles that are influential in their research community (i.e., articles with high bibliographic score) or articles that pertain to the specifics of their work (e.g., a particular article their work builds on or a specific method they use). Intuitively, in the former case, the model should put more weight on the bibliographic score, while in the latter case a higher weight on semantic score would be expected.

Semantic module. Similarly to Dai et al. (2019), we use bidirectional long short-term memory (Bi-LSTM) cells (Hochreiter and Schmidhuber, 1997) to represent citation context, but also to represent both cited article and global information from the citing article. Text fed to the module is segmented and tokenized using spaCy.[3] Target citation and other citations are masked with `TARGETCIT` and `OTHERCIT` placeholders, respectively. All three textual inputs are passed through the same two layers: LSTM and attention layer.

Let n be the total number of tokens in the input sequence $s = (t_1, \ldots, t_n)$. Each token t_i is mapped to a d_e-dimensional embedding vector $\mathbf{x}_i \in \mathcal{R}^{d_e}$ to generate a sequence $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$; we use pretrained embeddings from (Bhagavatula et al., 2018). Sequence $\mathbf{x}$ is then passed through a Bi-LSTM layer with hidden state dimension of d_h, where output $\mathbf{h}_i$ at each step i is formed by concatenating backward and forward hidden states: $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]$, $\mathbf{h}_i \in \mathcal{R}^{2d_h}$. The hidden states of input sequence s are passed through an additive attention layer (Bahdanau et al., 2015) to produce the final sequence embedding $\mathbf{z}_s$. Given input query vector $\mathbf{q}$ and hidden state vector $\mathbf{h}_i$, attention score for each step i is:

$$a_i = \mathbf{v} \cdot \tanh(W \cdot [\mathbf{q}; \mathbf{h}_i]) \quad (1)$$

where $\mathbf{v}$ and W are model parameters. Attention scores are normalized, applied to corresponding hidden states and summed to produce the final sequence embedding $\mathbf{z}_s$. A different query vector $\mathbf{q}$ is used depending on the input type. For citation context, we use the hidden state corresponding to

[3] https://spacy.io

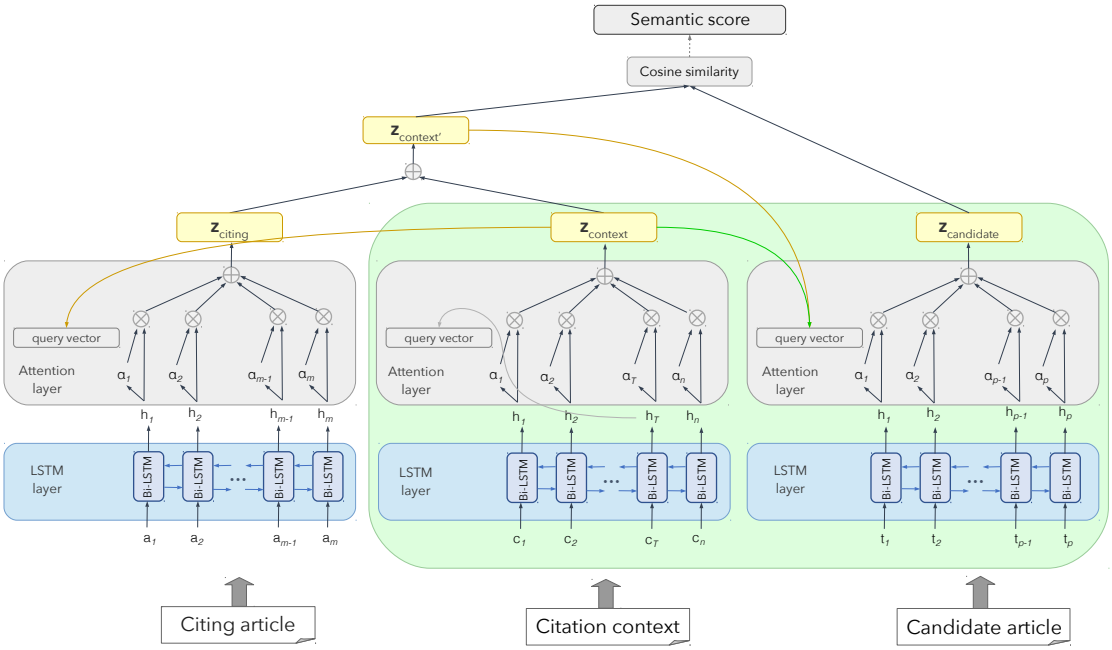

Figure 1: The semantic module of the local citation recommendation model. All text sequences pass through the same layers, with differences in attention query usage (indicated with arrows). The green oval corresponds to the *Con module variants, in which global information is not included.

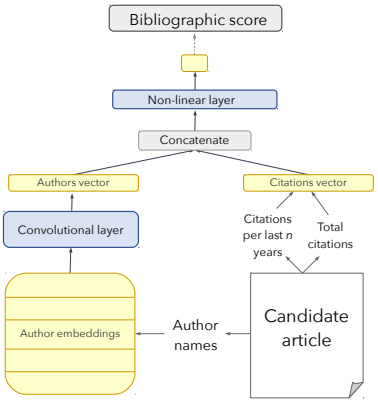

Figure 2: Architecture of the bibliographic module of the local citation recommendation model.

citation's placeholder ($\mathbf{h}_T$), so to focus context representation on the specific citation being predicted. For citing article's text, we use the final sequence embedding of the context ($\mathbf{z}_{\text{context}}$), while for candidate article's text we use the sum of citing article and context embeddings ($\mathbf{z}_{\text{citing}} + \mathbf{z}_{\text{context}}$). This allows the model to focus on context-specific information in both the citing and cited articles. More specifically, by using citation's placeholder hidden state as a query vector for calculating attention scores over citation context's tokens, we expect the model to focus on tokens in the context that are relevant for obtaining the embedding of the citation placeholder. Similarly, by using citation context embedding as a query for citing (or cited) article's text, the model focuses on tokens in the text that are relevant for given citation context, since article's text describes various aspects of an article and not all of them need to be equally relevant for the context at hand.

Given context c and candidate article p, semantic score scoring function $s_{\text{sem}}(c, p)$ is defined as cosine similarity between the enhanced context embedding and candidate article's embedding.

Bibliographic module. When the semantic context admits a number of citations, we assume the authors would prefer to cite articles that are well-known in the community. This is captured by the bibliographic module, which takes the authors' names and citation counts of candidate article p as input and produces a single bibliographic score.

Similarly to Ebesu and Fang (2017), we represent author names as embeddings. A sequence of author names $a = (a_1, \ldots, a_m)$ is first transformed into a sequence of author embeddings $\mathbf{a}_e = (\mathbf{a}_{e1}, ..., \mathbf{a}_{em})$. The sequence $\mathbf{a}_e$ is then passed through a convolutional layer followed by max-pooling and non-linear transformation to produce the final author embedding, which is then concatenated with article's total citation count and citation counts in last y years. Lastly, the entire vec-

Dataset	Train	Val	Test	Papers
ACL-ARC	30,390	9,381	9,585	19,711
RefSeer	3,521,582	124,911	126,593	624,957

Table 1: Dataset statistics (num. of contexts and papers)

tor is passed through non-linear layer to produce the bibliometric score $s_{\text{bib}}(p)$.

Final recommendation score. The final recommendation score $s_{\text{fin}}(c, p)$ is calculated as a weighted sum of the scores $s_{\text{sem}}(c, p)$ and $s_{\text{bib}}(p)$. Score weights are obtained by passing a context embedding $\mathbf{z}_{\text{context}}$ through a non-linear layer with two values at the output.

Loss function. We use the triplet loss to maximize recommendation score for true context-article pairs and minimize it for false pairs. The training set contains triplets (c, p_+, p_-), where c is the context, p_+ cited article, and p_- article not cited in the context. When sampling articles, we apply time-based filtering for choosing negative instances in triplets, i.e., we only consider articles published before the citing article. The loss function for a given scoring function $s(c, p_*)$ is defined as:

$$\mathcal{L}_s = \max(0, s(c, p_-) - s(c, p_+) + m) \quad (2)$$

where $s(c, p_*)$ is the recommendation score for article p_* and m is a margin used to enhance the difference between scores. We use the sum of three losses as the overall loss function, $\mathcal{L} = \mathcal{L}_{\text{sem}} + \mathcal{L}_{\text{bib}} + \mathcal{L}_{\text{fin}}$, where $\mathcal{L}_{\text{sem}}$, $\mathcal{L}_{\text{bib}}$, and $\mathcal{L}_{\text{fin}}$ correspond to semantic, bibliographic, and final recommendation score, respectively.[4]

4 Evaluation

We evaluate our model on two datasets: RefSeer (Huang et al., 2015) and ACL-ARC (Bird et al., 2008), both of which were used in work on local CR (Ebesu and Fang, 2017; Dai et al., 2019). Dataset statistics are given in Table 1. RefSeer dataset splits are subsets of those used in (Ebesu and Fang, 2017), while ACL-ARC splits are ours, as we were unable to obtain the version used in (Dai et al., 2019). Details on hyperparameters, prefiltering, and training are given in the appendix.[5]

RefSeer.[6] This dataset comprises articles and citation contexts from various engineering domains. Citation contexts are excerpts of text spanning 200 characters before and after citation. We filter out about 230k articles whose title and abstract is shorter than 100 characters, presumably due to parsing errors. As for most articles the publication year is missing, we use only the total number of citations in the bibliographic module and do not apply time-based filtering for the triplet loss. We adopt the same data splits as in (Ebesu and Fang, 2017), however, due to filtering, our numbers differ.

ACL-ARC.[7] This dataset contains articles published at ACL venues. We use the version processed using ParsCit (Councill et al., 2008), with citation contexts of 600 characters before and after citation. To investigate the effect of different context sizes on model performance, we use two dataset variants: ACL-600 (citation context size of $\pm$600 characters) and ACL-200 (citation context size of $\pm$200 characters, i.e., the same as for RefSeer). As each article's ID contains a year of publication, we use this information for citations counts features and for time-based filtering when constructing triplets. We use time-based data splits: contexts from years 2009 to 2013 are in train, from 2014 in validation, and from 2015 in test set.

Results. We compare our model against BM25 (Robertson and Walker, 1994) on both datasets and Neural Citation Network (NCN) (Ebesu and Fang, 2017) on the RefSeer dataset.[8] We fine-tuned BM25 on validation part of the ACL-600 dataset by performing a grid search over a range of values for parameters b and k_1. We then trained the model on all three datasets with the best performing parameters. Details on BM25 fine-tuning are given in the Appendix.

In addition to the complete model, we also test four of its variants: (1) SemCon – only semantic score, with standard context representation (without citing article's text), (2) SemEnh – only semantic score, with enhanced context representation, (3) DualCon – both scores, without enhanced context, and (4) DualEnh – both scores, enhanced context.

[4] We tried taking only $\mathcal{L}_{\text{fin}}$ as the overall loss, but obtained worse results compared to the composite loss function.

[5] Both our code and dataset splits are publicly available: https://github.com/zoranmedic/DualLCR.

[6] https://github.com/chbrown/refseer

[7] acl-arc.comp.nus.edu.sg

[8] Unfortunately, we were not able to replicate the NCN results of Ebesu and Fang (2017) on the RefSeer dataset, so we include the result they reported (albeit on a slightly larger dataset of 148,927 contexts) and, consequently, do not apply NCN to ACL datasets. Färber et al. (2020) also reported they were unable to replicate the results of the NCN.

	ACL-600		ACL-200		RefSeer	
Model	R@10	MRR	R@10	MRR	R@10	MRR
BM25	.095	.049	.095	.049	.090	.050
NCN	–	–	–	–	*.291*	*.267*
SemCon	.568	.306	.537	.291	.340	.166
SemEnh	.553	.290	.546	.285	.445	.216
DualCon-s	.654	.322	.693	.340	.363	.185
DualCon-ws	.689	**.368****	.647	.335	.406	.206
DualEnh-s	.662	.315	**.716**	.341	.437	.230
DualEnh-ws	**.699**	.357	.703*	**.366***	**.534***	**.280***

Table 2: Results on RefSeer and two ACL-ARC dataset variants for baselines and variants of the proposed model with (*-ws*) and without (*-s*) final score weighting. ** and * indicate statistically significant difference between DualEnh-ws and DualCon-ws for $p<0.05$ and $p<0.01$, respectively (two-sided t-test for MRR and two-proportion z-test for R@10).

Additionally, we evaluate Dual model versions with and without weighting of the two scores in the final recommendation score. In line with most previous work (Ebesu and Fang, 2017; Dai et al., 2019), we report recall at 10 (R@10) and MRR (Voorhees et al., 1999) (calculated on top 10 recommendations) on test sets. Candidate articles were obtained as top n articles retrieved by BM25 (n=2048 for RefSeer as in Ebesu and Fang (2017), and n=2000 for ACL variants), with cited article included in top n articles, if not retrieved initially.

Results are given in Table 2. Overall, dual scoring models achieve the best results on all three datasets, while models with weighed sum scoring achieve better results than those without (except for R@10 on ACL-200). On RefSeer, the best results achieved with DualEnh model are better than those reported by Ebesu and Fang (2017) (although on a smaller test set and possibly different set of candidate papers). On two ACL-ARC variants, the Dual model performance varies. On ACL-600, R@10 is the highest with DualEnh, albeit without significant improvement compared to DualCon, while DualCon achieves the best MRR score. On the other hand, DualEnh models yield the best scores with both metrics on both ACL-200 and Refseer datasets. The differences with respect to context size are also observable with semantic models – SemCon outperforms SemEnh on ACL-600, on ACL-200 this gap is smaller, while on RefSeer SemEnh outperforms SemCon. Taken together, our results suggest that enhancing context representation with global information helps when citation contexts are smaller, but not when context are longer, as they seem to provide sufficient information for local CR.

5 Conclusion

We presented a model for local citation recommendation with citation contexts enhanced with global information, i.e., the text from citing article's title and abstract. Model shows improvements over a competitive model (Ebesu and Fang, 2017) on the RefSeer dataset and its variants on datasets with smaller citation context sizes. For future work, we plan to further investigate the differences in performance across different datasets.

Acknowledgments

The first author has been supported by a grant from the Croatian Science Foundation (HRZZ-DOK-2018-09).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Steven Bethard and Dan Jurafsky. 2010. Who Should I Cite: Learning Literature Search Models from Citation Behavior. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 609–618.

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)*, pages 238–251.

Steven Bird, Robert Dale, Bonnie J Dorr, Bryan Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R Radev, and Yee Fan Tan. 2008. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1755–1759. EUROPEAN LANGUAGE RESOURCES ASSOC-ELRA.

Isaac G Councill, C Lee Giles, and Min-Yen Kan. 2008. Parscit: an Open-source CRF Reference String Parsing Package. In *LREC*, volume 8, pages 661–667.

Tao Dai, Li Zhu, Yaxiong Wang, and Kathleen M Carley. 2019. Attentive Stacked Denoising Autoencoder with Bi-LSTM for Personalized Context-aware Citation Recommendation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

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*, pages 1093–1096.

Michael Färber, Timo Klein, and Joan Sigloch. 2020. Neural citation recommendation: A reproducibility study. In *Proceedings of the 10th International Workshop on Bibliometric-enhanced Information Retrieval*, pages 66–74.

Michael Färber and Adam Jatowt. 2020. Citation Recommendation: Approaches and Datasets. *International Journal on Digital Libraries*.

Lantian Guo, Xiaoyan Cai, Fei Hao, Dejun Mu, Changjian Fang, and Libin Yang. 2017. Exploiting Fine-grained Co-authorship for Personalized Citation Recommendation. *IEEE Access*, 5:12714–12725.

Qi He, Daniel Kifer, Jian Pei, Prasenjit Mitra, and C Lee Giles. 2011. Citation Recommendation without Author Supervision. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining*, pages 755–764.

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*, pages 421–430.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation*, 9(8):1735–1780.

Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C Lee Giles, and Lior Rokach. 2012. Recommending Citations: Translating Papers into References. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1910–1914.

Wenyi Huang, Zhaohui Wu, Chen Liang, Prasenjit Mitra, and C Lee Giles. 2015. A Neural Probabilistic Model for Context Based Citation Recommendation. In *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, pages 2404–2410.

Arif E. Jinha. 2010. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, 23(3):258–263.

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 & Development in Information Retrieval*, pages 807–816.

Xiang Ren, Jialu Liu, Xiao Yu, Urvashi Khandelwal, Quanquan Gu, Lidan Wang, and Jiawei Han. 2014. Cluscite: Effective Citation Recommendation by Information Network-based Clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 821–830.

Stephen E Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR'94*, pages 232–241. Springer.

Ellen M Voorhees et al. 1999. The TREC-8 Question Answering Track Report. In *Proceedings of TREC-8*, pages 77–82.

Mark Ware and Michael Mabe. 2015. The STM Report: An overview of scientific and scholarly journal publishing.

A Model Training Details

Below we provide details on model training on both datasets. Model hyperparameters are given in Table 3.

Hyperparameter	**ACL-ARC**	**RefSeer**
LSTM hidden size	100	100
author embedding size	50	50
author embeddings number	4563	20000
v dimension	200	200
W dimension	400×400	400×400
non-linearity	sigmoid	sigmoid
cnn kernel sizes	$[1, 2]$	$[1, 2]$
cnn out channels	$[50, 50]$	$[50, 50]$
cnn non-linearity	ReLU	ReLU
n years features	4	n/a
title+abstract token cutoff	200	200
epochs	10	3
optimizer	Adam	Adam
learning rate	0.001	0.001
betas range	$(0.9, 0.999)$	$(0.9, 0.999)$
loss margin (m)	0.3	0.3
queries per batch	6	30
triplets per query	50	10
predict batch size	300	300
validation triplets per query	10	10

Table 3: Model's hyperparameter setting used for both datasets

Dataset construction. When constructing the ACL-ARC dataset variants, we match articles cited in citation context with the corresponding article in the collection by matching on lower-cased titles. When prefiltering authors, we keep those with more than 3 citations in the ACL-ARC dataset, and those with more than 5 citations in the RefSeer dataset.

Experiments environment. We run the ACL-ARC experiments on NVidia GeForce GTX 1080 GPU, with training time of approximately 5 hours. The RefSeer dataset experiments were run on NVidia GeForce RTX 2080 GPU, taking approximately 27 hours for the training.

Validation loss per model. To ensure reproducibility, we provide the best achieved validation loss for each model on all three datasets in Table 4.

Model	ACL-600	ACL-200	RefSeer
SemCon	0.01295	0.01853	0.00800
SemEnh	0.01239	0.01089	0.00400
DualCon-s	0.00279	0.00401	0.00512
DualCon-ws	0.04940	0.05080	0.12409
DualEnh-s	0.03469	0.00334	0.11590
DualEnh-ws	0.03635	0.03885	0.11578

Table 4: Best validation loss for each model and dataset

BM25 fine-tuning. For fine-tuning BM25 on ACL-600 dataset, we consider a range of values for parameters b and k_1. Parameter b was tested with in the following range: $[0.25, 0.5, 0.75]$, while k_1 was tested in the range: $[0.5, 1.5, 2.5]$. Table 5 contains results obtained on validation set for each combination of parameters.

Combination	R@10	MRR
$b = 0.25, k_1 = 0.5$	0.02878	0.01345
$b = 0.25, k_1 = 1.5$	0.02036	0.01036
$b = 0.25, k_1 = 2.5$	0.01418	0.00729
$b = 0.5, k_1 = 0.5$	0.06972	0.03557
$b = 0.5, k_1 = 1.5$	0.06407	0.03375
$b = 0.5, k_1 = 2.5$	0.04648	0.02545
$b = 0.75, k_1 = 0.5$	0.12824	0.06437
$b = 0.75, k_1 = 1.5$	**0.14721**	**0.07880**
$b = 0.75, k_1 = 2.5$	0.13570	0.07162

Table 5: BM25 grid search results on validation set of ACL-600 dataset.