Neural Citation Recommendation: A Reproducibility Study

Michael Färber, Timo Klein, and Joan Sigloch

Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
michael.faerber@kit.edu
{ufpzp,uzoqf}@student.kit.edu

Abstract. Context-aware citation recommendation is used to overcome the process of manually searching for relevant citations by automatically recommending suitable papers as citations for a specified input text. In this paper, we examine the reproducibility of a state-of-the-art approach to context-aware citation recommendation, namely the neural citation network (NCN) by Ebesu and Fang [1]. We re-implement the network and run evaluations on both RefSeer, the originally used data set, and arXiv CS, as an additional data set. We provide insights on how the different hyperparameters of the neural network affect the model performance of the NCN and thus can be used to improve the model's performance. In this way, we contribute to making citation recommendation approaches and their evaluations more transparent and creating more effective neural network-based models in the future.

Keywords: recommender systems, bibliometrics, citation context

1 Motivation

Citing sources is an essential part of academia to guarantee transparency and truthfulness. However, the process of finding relevant and appropriate citations is becoming increasingly time-consuming and difficult due to the sheer amount of new literature published every year [1]. *Citation recommendation* [2] has been proposed to overcome this issue. This task refers to the idea of generating a ranked list of potentially suitable citations in an automated way, thus facilitating the process of choosing correct citations.

According to He et al. [3], there are two types of citation recommendation tasks, namely *global citation recommendation* and *local citation recommendation*. The former is used to propose candidates for the bibliography of a given scientific manuscript that does not yet have a bibliography. Local citation recommendation, on the other hand, proposes candidates for a given citation placeholder (e.g., "[1]") located in the written text of a scientific document. In

order to generate recommendations, the text surrounding the placeholder, often referred to as the *citation context*, is used as an input into the recommender system. The output consists of a ranked list containing candidates for the query placeholder.

In recent years, several approaches to global and local citation recommendation have been proposed [2]. In this paper, we analyze the reproducibility of one specific local citation recommendation approach, namely the *neural citation network* (NCN) by Ebesu and Fang [1]. We chose this approach due to its currentness, its promising results on a large data set, and its wide acceptance in the scientific community (based on citation counts). Note that we were unable to run the source code published online by Ebesu and Fang. Furthermore, the Python version used by Ebesu and Fang is outdated. Thus, after re-implementing the network, we used both the original data set and another data set for training and evaluating the NCN in order to examine its performance under varying circumstances.

Overall, we make the following contributions in this paper:

1. We re-implement the NCN by Ebesu and Fang [1], a state-of-the-art approach to local citation recommendation.
2. We run extensive experiments based on the NCN using the original data set *RefSeer* and *arXiv CS* as a further data set.
3. We analyze the evaluation results and give noteworthy conclusions for the future development of local citation recommendation approaches.

The rest of this paper is structured as follows: We give an overview of the NCN architecture in Sec. 2. In Sec. 3, we present our experimental setup and the evaluation results. We conclude in Sec. 4.

2 The Neural Citation Network

The NCN proposed by Ebesu et al. [1] consists of an encoder-decoder model coupled with an attention mechanism (see Fig. 1).

Encoders. Encoders are deployed as part of the NCN in order to turn the raw citation context and the citing/cited authors' names into feature tensors holding important information about the context and the authors, respectively.

1. **Context encoder.** The part of the NCN that is responsible for encoding the citation context is a *time-delay neural network* (TDNN) introduced by Collobert et al. [4]. It allows multiple forward propagations through the network at once, leading to all feature maps being calculated in parallel. The TDNN used by Ebesu and Fang consists of a convolutional layer followed by both a pooling layer and a fully connected layer.
2. **Author encoder.** In order to include author information when generating citation recommendations, the NCN comprises an author encoder, which uses the same architecture as the context encoder (outlined above). It is separately applied to (1) the embeddings of the authors' names $\boldsymbol{A}^q$ of the document from which the query context originated as well as (2) the embeddings of the authors' names $\boldsymbol{A}^d$ of all documents in the database. The

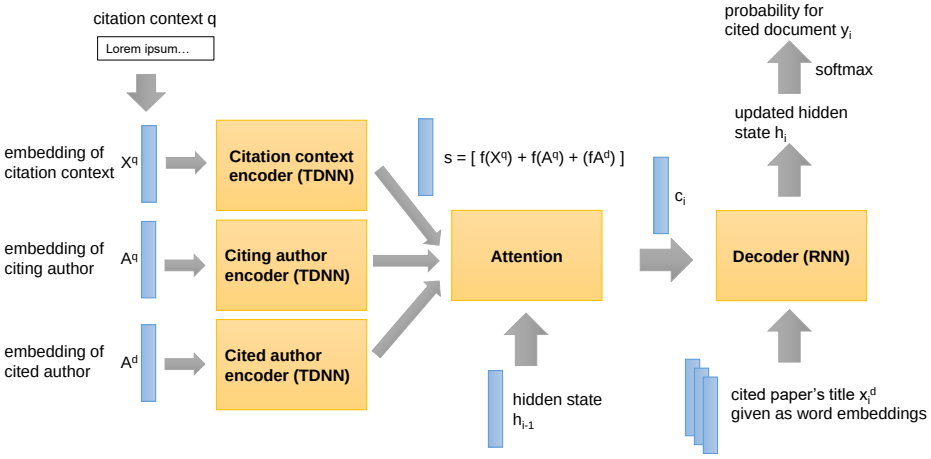

Fig. 1: Architecture of the neural citation network (NCN).

author encoder is applied multiple times using TDNNs with varying region filter sizes in the convolutional layer.

The final representation which results from applying the context encoder and author encoders is denoted as

$$\boldsymbol{s} = [f(\boldsymbol{X}^q) \oplus f(\boldsymbol{A}^q) \oplus f(\boldsymbol{A}^d)],$$

with a given citation context representation $\boldsymbol{X}^q$.

Decoder. The NCN's decoder is a recurrent neural network (RNN) that makes use of the gated recurrent unit (GRU) [5] as a gating mechanism as well as the attention mechanism [6]. It is applied to the title of every document that can be used as a citation for the query citation context.[2] The purpose of the decoder is to generate scores for every document in the database indicating its suitability as a citation for the given query context. The scores can ultimately be used to generate citation recommendations for the query context.

Attention mechanism. The NCN makes use of the attention mechanism originally introduced by Bahdanau et al. [6]. With the help of attention, the encodings $\boldsymbol{s}_j$ that originate from the context and author encoders are given weights dependent on the decoder output $\boldsymbol{h}_{i-1}$ for the word prior to i. The result is a context vector $\boldsymbol{c}_i$ which is made up of a weighted sum of the encoder outputs $\boldsymbol{s}_j$ in accordance to their relevance. Attention is used to put emphasis on encodings that are particularly important for the current time step. The attention mechanism is implemented as a feed-forward neural network that concludes with a softmax layer converting attention vectors $\boldsymbol{a}_{ij}$ into attention scores α_{ij}. These indicate the importance of the encoder output $\boldsymbol{s}_j$ for the i^{th} word in the title of the document currently being decoded. To illustrate, in

[2] For very large databases, a pre-selection algorithm may make sense to save computing time. See Section 3.2 for further information.

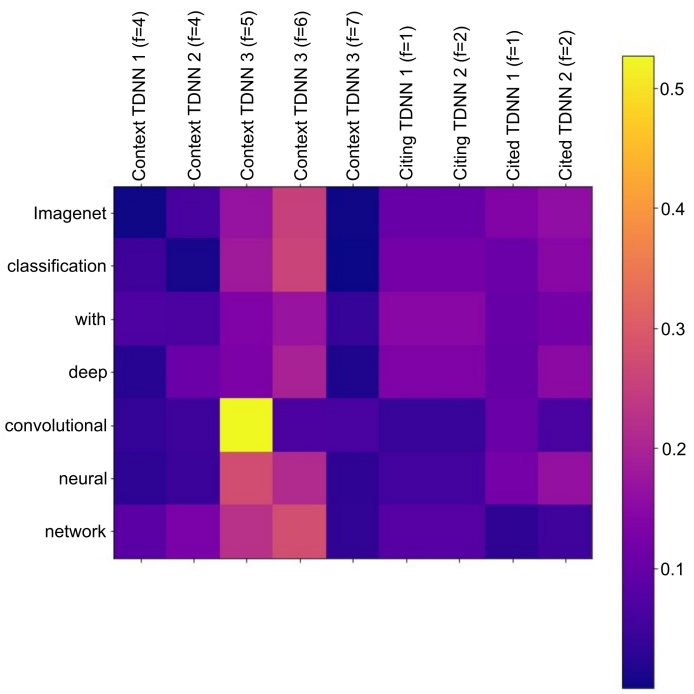

Fig. 2: Illustrative example of attention weights.

Fig. 2, we visualize the matrix α_{ij} for the target sentence "Imagenet classification with deep convolutional neural networks" by the well-known authors "Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton" after the sequence was tokenized and preprocessed. The context for this example was set to "Neural networks are really cool, especially if they are convolutional" with the authors "Chuck Norris, Bruce Lee." This toy example visualizes how the decoder sensibly puts little emphasis on the citing authors as compared to the context and cited authors. The context vector $\boldsymbol{c}_i$ is determined for every word i in the document title.

3 Evaluation

3.1 Data Sets

We used two data sets in our evaluation.

1. **RefSeer.** Following Ebesu and Fang [1], we used RefSeer [7] as our first data set. Although we followed Ebesu and Fang's instructions on creating their evaluation data set, we were unable to generate the exact same data set based on the original RefSeer data as we were unable to find any information about citing authors within the data set, only cited authors. For comparison, we decided to randomly select 4.5 M out of the generated 14.9 M citation

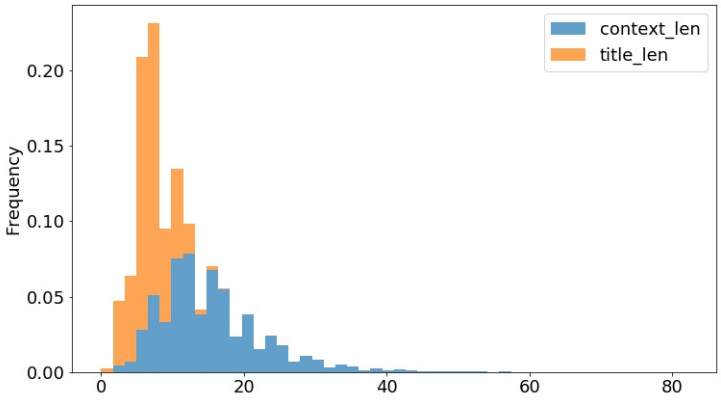

Fig. 3: Distribution of citation context and citation title lengths in the preprocessed arXiv CS data set.

contexts in order to end up with the same data set size as the one used by Ebesu and Fang. Note that the data set we reused did not contain author information of the citing papers. We thus expected poorer performance than that of the model published by Ebesu and Fang.

2. **arXiv CS.** We used the arXiv.org publications in the computer science domain as our second data set, as proposed by Färber et al. [8] for citation-based tasks. We cut off the citation contexts and citation titles at lengths of 100 and 30 words, respectively, to achieve a trade-off between model performance and training time (see Fig. 3). Overall, we used 502,353 pairs of citations and citation contexts. We chose this data set in order to obtain insights into how well our models perform under different circumstances than the ones presented by Ebesu and Fang. Thus, our paper is not only a replicability paper (with a focus on repeating prior experiments to see when the methods work) but also a reproducibility paper (repeating experiments in new contexts).

For model training and evaluation, we split the data sets into 80% training, 10% validation, and 10% test data sets and set a seed to ensure reproducibility.

3.2 Model Re-Implementation

We rebuilt the NCN from scratch. Our final code is available on GitHub.[3] We used PyTorch to reimplement the network, which was originally coded in TensorFlow version r0.11. We used the torchtext package to convert the data set into a suitable format for PyTorch and to facilitate the preprocessing

[3] See `https://github.com/X3N4/neural_citation`.

Table 1: Results of our replicability and reproducibility studies on the neural citation network (NCN) [1] using the recall@10 metric. We show the total number of trainable parameters ("# Param.") as an indicator of model complexity.

Model	**RefSeer**		**arXiv CS**	
	# Param.	**Recall@10**	**# Param.**	**Recall@10**
Embedding, conv filters, hidden: 64				
Ebesu & Fang [1]	7,890,916	0.0929	*7,919,716*	*0.1637*
Batch size: 32	7,890,916	0.0876	7,919,716	0.1662
Vocab size: 30k	11,740,916	**0.0945**	11,769,716	0.1672
Filters: [4,4,5,6,7]	8,009,828	0.0916	8,038,628	0.1661
GRU layers: 1	7,865,956	0.0914	7,894,756	0.1592
GRU layers: 3	7,915,876	0.0846	7,944,676	0.1643
Combined improvements	11,884,788	0.0925	11,913,588	0.1621
Embedding, conv filters, hidden: 128				
Size: 128	16,138,660	0.0878	16,253,604	0.1748
Filters: [4,4,5,6,7]	16,614,052	0.0849	16,728,996	**0.1797**
Impr. Filters, Batch size:32	16,614,052	0.0835	16,728,996	0.1637
Vocab size: 30k	23,828,660	0.0911	23,943,604	0.1705
Combined improvements	24,304,052	0.0871	24,518,068	0.1695
Embedding, conv filters, hidden: 256				
Batch size: 32	33,764,644	0.0877	34,223,908	0.1268

steps. Furthermore, we used the SpaCy library in combination with torchtext to tokenize the data set. After lemmatizing the data and removing stopwords using the combined SpaCy and nltk stopword corpora, we numericalized the data set using a vocabulary size of 20,000 tokens for citation contexts, citation titles, and authors. To facilitate propagating batches through the network, we made use of the bucketing technique that Ebesu and Fang used as well. Like Ebesu and Fang, we further use the BM25 ranking function in the decoder part of the network to preselect citation titles for a given citation context.

3.3 Evaluation Results

Citation recommendation approaches are difficult to evaluate, as the citation provided by the original authors cannot be seen as the unequivocal ground truth. Therefore, we did not consider ranking metrics but solely *recall@k* as our evaluation metric. Table 1 shows the evaluation results.

RefSeer. We were unable to run our code on exactly the same data set as Ebesu and Fang did, and our model for RefSeer does not include citing authors' information (see Sec. 3.1), leading to a slightly different number of parameters. Presumably due to the missing citing author information, our results are worse than the ones reported by Ebesu and Fang (namely, recall@10 of 0.0929 instead of around 0.29). Overall, all of the recall@10 values were in a similar range. However, using other setups than the one proposed by Ebesu and Fang seems promising.

arXiv CS. We evaluated our trained models on the first 20,000 of the 50,235 test examples, which significantly reduced the evaluation running time and allowed us to perform detailed ablation studies.

By applying the hyperparameters used by Ebesu and Fang, our reimplemented NCN yielded a recall@10 of 0.1637, as compared to 0.29 in the original paper. Thus, we were unable to replicate the performance of the original model. We hypothesize that this is a result of our significantly smaller dataset, which comprised only 9.44% of the original paper's training examples (401,882 examples compared to 4,258,383 in the original paper). In order to tune performance, we used differing hyperparameter settings and evaluated our model after every modification. Our changes included the use of different vocabulary sizes when preprocessing the data set as well as varying batch and embedding sizes when propagating data through the network. We also altered the number of filters in the convolutional layer of the TDNN encoder and the number of GRU layers in the RNN decoder. Table 1 shows that the best configuration achieved a 9.77% improvement compared to Ebesu and Fang's hyperparameter values (recall@10 of 0.1797 vs. 0.1637). While the NCN's performance increases with larger capacity in general, this effect only persists up to a certain size.[4] In particular, enlarging the embedding size past 128 dimensions and increasing the vocabulary to more than 20,000 tokens did not guarantee an improved recall@10 value. We suspect this to be the result of our small data set, as compared to the model's increased capacity [9].

In addition to experimenting with various architectural changes, we also tried different batch sizes. Masters et al. [10] showed that training with smaller mini-batches can lead to improved test performance. However, we were unable to replicate these results for our best configuration. For the larger NCN models, a decreased batch size instead led to inferior test performance. On the other hand, our enhanced filter region sizes for the TDNN context encoder consistently boosted the model's performance. At the same time, this modification is computationally cheap, in terms of both additional parameters and wall time, as the TDNN encoders run in parallel.

We observed during the evaluation runs that models with a lower validation loss generally achieved a better recall@10 value (given equal batch sizes). While this intuitively makes sense, as we use the loss function to re-rank the top titles, we can also find counterexamples.

3.4 Discussion

We believe that there is still room to improve the NCN, in terms of the model's hyperparameters and architecture. Our research shows that changing the filter lengths in the convolutional layer of the network's encoder leads to consistently better results. Further investigation into their effects on model improvement

[4] We use the term "model size" to refer to the embedding dimension, number of convolutional filters, and the GRU dimension. These parameters are set to the same value in most configurations.

may thus be rewarding. The original architecture only used Dropout [11] to regularize the network. It may be worthwhile to investigate other regularization techniques such as batch normalization [12] for convolutional layers or layer normalization [13] for recurrent layers.

We conclude that the NCN leads to reasonable results even when applied to a smaller data set, like the arXiv CS subset used in our paper. We believe a major reason for not being able to achieve similar performance results on another data set (arXiv CS) was the significantly smaller size of training examples [9]. Thus, for the future, it might be more important to use large data sets than to further tune model hyperparameters in order to obtain better recall@10 scores.

4 Conclusion

For this paper, we re-implemented the neural citation network [1] for citation recommendation and ran evaluations on both RefSeer, the originally used data set, and arXiv CS, as the second evaluation data set. We were unable to achieve the same model performance as Ebesu and Fang did. However, we provided insights on how the different hyperparameters can affect the NCN's model performance and how these insights can be used to further improve the model. In this way, we exemplified how to make citation recommendation approaches and their evaluations more transparent facilitating the creation of more effective models in the future.

References

1. Ebesu, T., Fang, Y.: Neural Citation Network for Context-Aware Citation Recommendation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR'17 (2017) 1093–1096
2. Färber, M., Jatowt, A.: Citation Recommendation: Approaches and Datasets. CoRR **abs/2002.06961** (2020)
3. He, Q., Pei, J., Kifer, D., Mitra, P., Giles, L.: Context-aware Citation Recommendation. In: Proceedings of the 19th International Conference on World Wide Web. WWW '10 (2010) 421–430
4. Collobert, R., Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In: Proceedings of the 25th International Conference on Machine Learning. ICML'08 (2008) 160–167
5. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP'14 (2014) 1724–1734
6. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. In: Proceedings of the 3rd International Conference on Learning Representations. ICLR'15 (2015)
7. Huang, W., Wu, Z., Mitra, P., Giles, C.L.: RefSeer: A citation recommendation system. In: Proceedings of the 2014 IEEE/ACM Joint Conference on Digital Libraries. JCDL'14 (2014) 371–374

8. Färber, M., Thiemann, A., Jatowt, A.: A High-Quality Gold Standard for Citation-based Tasks. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation. LREC'18 (2018)
9. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In: Proceedings of the 2017 IEEE International Conference on Computer Vision. ICCV'17 (2017) 843–852
10. Masters, D., Luschi, C.: Revisiting Small Batch Training for Deep Neural Networks. CoRR **abs/1804.07612** (2018)
11. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research **15**(1) (2014) 1929–1958
12. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the 32nd International Conference on Machine Learning. ICML'15 (2015) 448–456
13. Ba, L.J., Kiros, J.R., Hinton, G.E.: Layer Normalization. CoRR **abs/1607.06450** (2016)