Local Citation Recommendation - Towards The Perfect Prefetching : A Comparative Analysis of Encoding Techniques for Citation Retrieval

Yael Einy
yaeleiny@mail.tau.ac.il

Tamir Sadovsky
tamirs1@mail.tau.ac.il

Ron Nezer
ronnezer@mail.tau.ac.il

Abstract

The goal of local citation recommendation is to recommend a missing reference from the local citation context and optionally also from the global context.The common Local Citation Recommendation aims to suggest missing references within the context of a scholarly paper's citations. To balance the tradeoff between speed and accuracy of citation recommendation in the context of a large-scale paper database, a viable approach is to first prefetch a limited number of relevant documents using efficient ranking methods and then to perform a fine-grained reranking using more sophisticated models. As BM25 and tf-idf are strong and fast prefetching approaches, recent efforts have concentrated only on the reranking phase. Hence, we explore the prefetching step - with nearest neighbor search among text embeddings, constructed by different LLM's encoders (Llama2 [26] , Bert [4] and SciBert [1]). In addition, we have created an enriched data-set, desinged for the task. Using our enriched data-set, when coupled with a SciBERT reranker, our encoders achieves impressive prefetch recall for a given number of candidates to be re-ranked.By training the baseline model [7] on our improved data-set, we attained state-of-the-art results for this task.

The source code for the project can be found at the following GitHub repository: Local Citation Recommendation - Towards The Perfect Prefetching.

1 Introduction

The digital age has ushered in a surfeit of scientific articles, with the publication record and the scientific vocabulary witnessing exponential growth in both the publication record [12] and the underlying vocabulary [10]. This has made the process of literature discovery increasingly intricate. Citation recommendation, where an uncited query text is the input and a potentially citable paper is the output, presents a viable solution to this burgeoning challenge. While global citation recommendation relies on the title and abstract of a paper as the query [25, 20, 2], local citation recommendation hinges on two primary contexts [9, 11, 13, 18]: 1) the local context - the text neighboring the citation placeholder; 2) the global context-the title and abstract of the paper making the citation. The local citation recommendation's objective is to pinpoint the paper that was originally cited at the placeholder in the local context. Our study delves deep into improving local citation recommendation, using advanced encoding methods and expanded data.
In designing an efficient local citation recommendation system, it is pivotal to strike a harmonious balance between speed and accuracy, especially when the system has to navigate through vast databases comprising millions of scientific papers. This equilibrium is traditionally achieved via a dual-stage approach:

1. A fast prefetching model first retrieves a set of candidate papers from the database;

2. A more sophisticated model then performs a fine-grained analysis of scoring candidate papers and reordering them to result in a ranked list of recommendations.

Historically, simple methodologies like TF-IDF [21] and BM25 [22] have taken center stage in the prefetching process. They operated in isolation, devoid of fine-tuning or integration into the overall performance evaluation.

Our research proposes a nuanced paradigm for the prefetching step of local citation recommendation. In our prefetching step, we channel the power of text embeddings, harnessing different encoding methods, with nearest neighbor search among the created text embeddings, to create an adept retrieval system. Our study demonstrates that by integrating our encoders with a SciBERT reranker [1], we achieve commendable prefetch recall.

Moreover, we employ an enriched data-set, paving the way for extensive papers embeddings.The dataset improves the prediction's recall. We have done a vast set of experiments and show the performance of our methods using our data-set and the baseline's [7] data-set. In summation, our contributions are threefold:

1. We introduce an avant-garde retrieval system that places paramount emphasis on the encoding techniques during the prefetching stage, subsequently pairing it with a pre-trauned SciBERT reranker.

2. We have created an expanded data-set for improving the prefetch recall.

2 Related Works

Local citation recommendation has undergone significant evolution, witnessing several methodologies and approaches. Initial methods, like the one proposed by He et al. [9], modeled the relevance between the query and each candidate citation through a non-parametric probabilistic model. This was followed by embedding-based solutions [15, 6], leveraging cosine or Euclidean distance metrics between embeddings to capture the similarity between queries and targets.

More recent advancements led Jeong et al. [13] to introduce the BERT-GCN model. This model combined the strengths of Graph Convolutional Networks [14] (GCN) and BERT [4] to compute embeddings from both citation graphs and query contexts. However, its practicality was hindered due to scalability challenges, particularly when dealing with expansive paper databases.

While some recent studies [18, 3, 5, 17] have adopted the prefetching-reranking strategy to address scalability, their prefetch components (like BM25 or TF-IDF) were mostly used for data-set creation, leading to artificially enhanced performance metrics. Such idealized prefetching models are unrealistic in real-world scenarios, creating a chasm between theoretical and practical performance.

The necessity for supervised citation recommendation techniques brings forth another challenge: the acquisition of substantial labeled data-sets. Parsing the complete texts of papers to extract local contexts and find citations that also exist within the data-set is an intricate task. Existing data-sets like ACL-200 [18] and FullTextPeerRead [13] are relatively small, while larger ones like RefSeer, mentioned in Medi ́c and Snajder [18], are outdated. Further, data-sets such as unarXive [23], while large, have their challenges in terms of structured parsing and citation context quality.

A paradigm shift in this domain was brought about by the baseline [7] work titled "Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-based Reranking". This paper proposed a two-step methodology: initially prefetching documents using text embeddings constructed by a hierarchical attention (HAtten) network and then rerannking by a fine-tuned SciBERT reranker. This methodology not only improved prefetch recall rates but also reduced the number of candidates needed for reranking, demonstrating state-of-the-art performance across diverse citation recommendation data-sets. Our follow-up work replaces the words-embeddings created the Hatten encoder of the baseline [7], by embeddings constructed by different LLM's encoders (Llama2 [26] , BERT [4] and SciBERT [1]).

3 Data Collection and Enrichment

Building upon the original data-set, which was derived from arXiv papers contained in S2ORC, we expanded and enriched the data-set of the baseline paper[7] using the Semantic Scholar API to provide a more comprehensive view of each paper. Here's a breakdown of the data collection and enrichment process:

1. Baseline's data-set Retrieval: We began with the papers.json, a data-set created by our the baseline paper [7], which contained only the title and abstract for each article. A significant challenge with expanding this data-set was that the baseline data-set did not possess unique identifiers or paper IDs for the articles.

2. API Key Request and Paper ID Mapping: To facilitate faster and multiple requests, we obtained an API key from Semantic Scholar.To overcome the lack of paper IDs in the baseline, we employed the search API of Semantic Scholar. By using the titles of the articles as queries, we were able to fetch the unique paper IDs for each article. This step was piv-

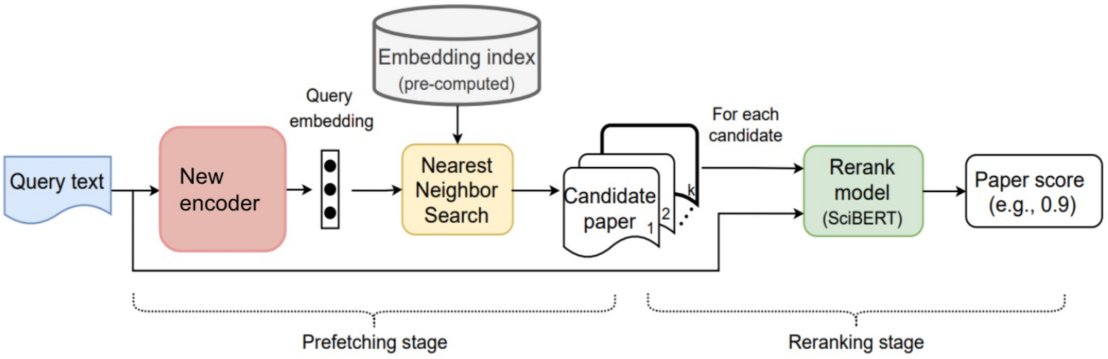


Figure 1: Overview of our two-stage local citation recommendation pipeline.

otal as it created a mapping between the title and the unique ID, making subsequent data enrichment steps feasible and accurate.

3. Metadata Retrieval: With the paper IDs in hand, we ran a process to fetch additional metadata for each article. The following fields were retrieved and appended to our data-set.

    - Year: The publication year of the paper.
    - Authors: List of authors who contributed to the paper.
    - Fields of Study: Domains or areas the paper pertains to.
    - isOpenAccess: A boolean flag indicating if the paper is open access.
    - isPublisherLicensed: A boolean flag indicating if the paper is licensed by the publisher.
    - numCitedBy: Number of citations the paper has received.
    - numCiting: Number of citations made by the paper.
    - s2FieldsOfStudy: Semantic Scholar's classification of the paper's fields of study.
    - Topics: The main topics or themes of the paper.
    - Venue: The publication or conference where the paper was presented or published.

4. Database Update: we updated the database.

By integrating this enriched data, we aimed to provide a more holistic representation of each paper, enhancing the capabilities of the local citation recommendation model.

Analyzing the Impact of Enriched Data on Model Performance After enriching our data-set, we proceeded to re-run our learning process. The primary objective was to gauge the impact of the additional data on the baseline's[7] performance. **Initial Observations And Model Adaption** To our initial surprise, the enriched data-set did not yield improved results compared to the previous version. We realized that data truncation, designed for the new data-set, should be done. We modified the truncation logic of the baseline model [7] to accommodate longer data inputs. This change significantly increased the runtime, which was a trade-off we were willing to make for potentially enhanced performance.

After accommodating the more extensive data, the results were still underwhelming, with no noticeable enhancement in the recommendation quality.

Data Overloading Hypothesis: A theory was posited that the influx of a vast amount of meta-data might be overwhelming the model and could introduce noise, rather than aiding it. **Refining the Data-set** In light of the data overloading hypothesis, We decided to retain only the most potentially impactful fields:

- Year: The publication year can provide temporal context, and discard articles which were published after the query article was introduced.

- Fields of Study: provides categorical context about the areas the paper relates to.

- Topics: Highlighting the main themes or keywords of the paper can offer contextual guidance for recommendation purposes.

By focusing on these fields, State-Of-The-Art results were successfully achieved (by the baseline model [7]).When creating this data-set, our aim was to strike a balance between essential context and avoiding data overload. The results from this refined approach not only reaffirmed the age-old adage - 'less can be more' but also marked a notable milestone in our experiments. By judiciously and selectively feeding the model with pertinent data, we achieved an enhancement in its efficacy that surpassed previous benchmarks. We report the entire results on section 6.

4 methods

4.1 Approach

Our two-stage telescope citation recommendation system is similar to that of [2, 7], composed of a fast prefetching model and a slower reranking model.

4.2 Prefetching Model

The prefetching model scores and ranks all papers in the database to fetch a rough initial subset of candidates. Similar approach has been taken by [8] [7]. The core of the prefetching model is a light-weight text encoder that efficiently computes the embeddings of queries and candidate documents. We designed some representation-focused models, by utilizing different LLM's encoders (Llama2 [26] , Bert [4] and SciBert [1]) for the task. Our encoders compute a query embedding for each input query, which is used to rank each candidate document according to the cosine similarity between the query embedding and the pre-computed document embedding. As shown in figure 2, the encoder processes each document or query in a two-level hierarchy, consisting of two components: a paragraph encoder and a document encoder.

Paragraph Encoder For each paragraph pi in the document, the paragraph encoder in left side of figure 2 takes as input the token sequence $p_i = [w_1, ..., w_{ni}]$, composed of n_i tokens (words) to output the paragraph embedding e_{pi} as a single vector. In order to incorporate positional information of the tokens, the paragraph encoder makes use of positional encoding. Contextual information is encoded with a single transformer encoder layer following the configuration in [27] Figure 2a. To obtain a single fixed-size embedding e_p from a variably sized paragraph, the paragraph encoder processes the output of the transformer encoder layer with a multi-head pooling layer [16] with trainable weights. Let $x_k \in^d$ be the output of the transformer encoder layer for token w_k in a paragraph p_i. For each head $j \in \{1, ..., n_head\}$ in the multi-head pooling layer, we first compute a value vector $v_k^j \in^{d/n_{heads}}$ as well as an attention score $\hat{a}_k^j \in$ associated with that value vector:

$$v_k^j = Linear_v^j(x_k), a_k^j = Linear_a^j(x_k)$$

$$\hat{a}_k^j = \frac{\exp a_k^j}{\sum_{n=1}^{n_{token}} \exp a_m^j}$$

where Linear() denotes a trainable linear transformation. The weighted value vector $\hat{v}^j$ then results from the sum across all value vectors weighed by their corresponding attention scores: $\hat{v}^j = \sum_{n=1}^{n_{paragraph}} \hat{a}_m^j v_j^m$ The final paragraph embedding ep is constructed from the weighted value vectors $\hat{v}^j$ of all heads by a ReLU activation [19] followed by a linear transformation:

$$e_p = Linear_p(ReLU(Concat(\hat{v}^1, ..., \hat{v}^{n_{head}})))$$

Document Encoder In order to encode documents with some fields of data, we treat each field as a paragraph. For a document of n_{par} paragraphs $d = [p_1, ..., p_{n_{\text{par}}}]$, we first compute the embeddings of all paragraphs p_i.

Not all fields and paragraphs are treated equally in our document encoder. To allow the document encoder to distinguish between fields, we introduce a paragraph type variable, which refers to the field type from which the paragraph originates. Each type is associated with a learnable type embedding that has the same dimension as the paragraph embedding. Inspired by the BERT model [4], we produce a type-aware paragraph embedding by adding the type embedding of the given paragraph to the corresponding paragraph embedding (Figure 2b). All type-aware paragraph embeddings are then fed into a transformer encoder layer followed by a multi-head pooling layer (of identical structures as the ones in the paragraph encoder), which then results in the final document embedding e_d.

Prefetched document candidates The prefetched document candidates are found by identifying the K nearest document embeddings to the query embedding in terms of cosine similarity. The

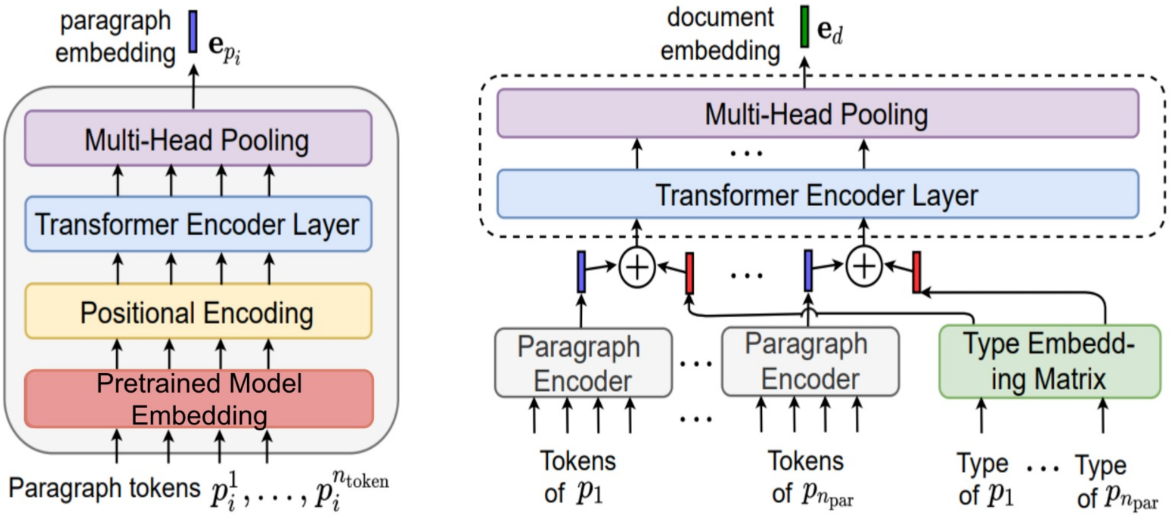


Figure 2

The pretrained model embeddings are generated using Llama2, BERT, and SciBERT.

ranking is performed using a brute-force nearest neighbor search among all document embeddings as shown in Figure 1.

4.3 Reranking Model

The reranking model performs a fine-grained comparison between a query q (consisting of a local and a global context) and each prefetched document candidate. The relevance scores of the candidates constitute the final output of our model. The reranker where designed by the our baseline model[7], and it is based on SciBERT [1], which is a a BERT model [4] trained on a large-scale corpus of scientific articles. The input of the SciBERT reranker has the following format: $[CLS]\, SentenceA\, [SEP]\, SentenceB$, where sentence A is the concatenation of the global context and the local context of the query, and sentence B is the concatenation of the title and the abstract of the candidate paper to be scored. The SciBERT encoded vector for the $[CLS]$ J token is then fed into a feed-forward network that outputs the relevance score $s \in [0, 1]$ provided via a sigmoid function.

4.4 Loss Function

We use a triplet loss for training. The triplet loss is based on the similarity $s(q, d)$ between the query q and a document d. For the prefetching step, $s(q, d)$ is given by the cosine similarity between the query embedding v_q and the document embedding v_d, both computed with our encoders. For the reranking step, $s(q, d)$ is given by the relevance score computed by the SciBERT reranker.

In order to maximize the relevance score between the query q and the cited document d^+ (the positive pair (q, d^+)) and to minimize the score between q and any non-cited document d^- (a negative pair (q, d^-)), we minimize the triplet loss:

$$L = \max[s(q, d^-) - s(q, d^+) + m, 0] \quad (1)$$

where the margin $m > 0$ sets the span over which the loss is sensitive to the similarity of negative pairs.

For fast convergence during training, it is important to select effective triplets for which L in Equation (1) is non-zero [24], which is particularly relevant for the prefetching model, since for each query there is only a single positive document but millions of negative documents (e.g., on the arXiv dataset). Therefore, we employ negative and positive mining strategies to train our encoders, described as follows.

5 Implementation Details And Experiments

5.1 Implementation Details

In the prefetching step, we chose a diverse range of pre-trained language models, as the foundation for our word embeddings. We made this selection based on their strong capability to effectively capture and represent textual information. Specifically, our choices were as follows.

Llama2 [26]: A large language model.
BERT [4]: A widely recognized pre-trained model.
SciBERT [1]: A domain-specific pre-trained model.

Throughout our experiments, we utilized embeddings from these pre-trained models, known for their proficiency in capturing contextual and semantic information within text. The intent behind using these models was to create a more comprehensive and context-aware representation of text to potentially yield superior results. There are 64 queries in a mini-batch, each of which was accompanied by 1 cited paper, 4 non-cited papers randomly sampled from the top $K_n = 100$ prefetched candidates, and 1 randomly sampled paper from the whole database, which allow us to do negative and positive mining with the mini-batch as described in Section 4.3. The new encoders checkpoint was updated every $N_{\text{iter}} = 5000$ training iterations.

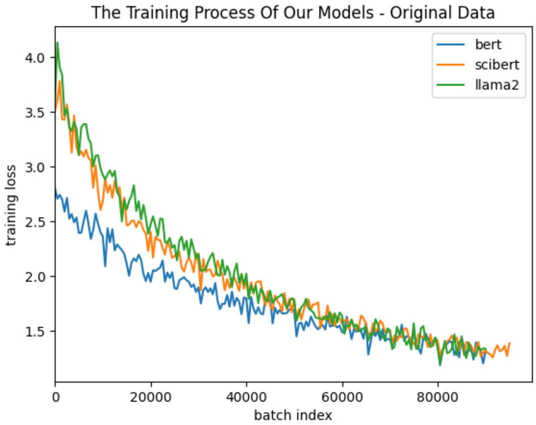
The learning rate was set to $\alpha = 1e^{-4}$ and the weight decay to $1e^{-5}$.

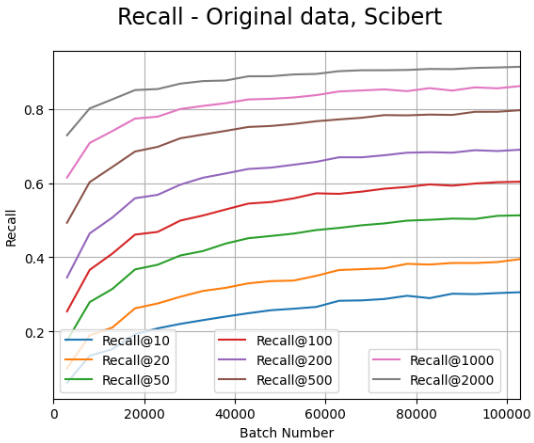
5.2 Evaluation Metrics

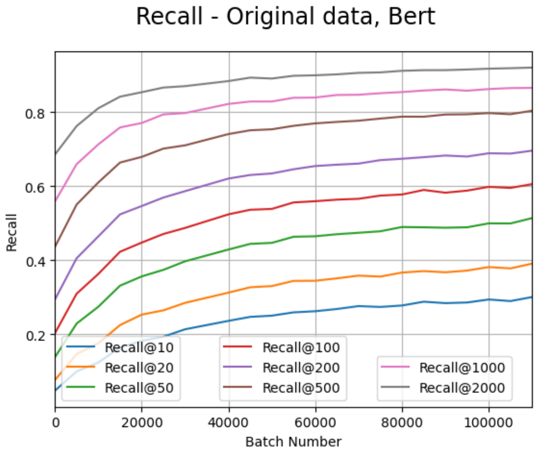
We evaluated the recommendation performance using the Recall@K (Recall for k, R@K for short). Here, K denotes the number of candidate papers considered during the prefetching stage. The primary goal of using R@K is to ascertain how many times a cited paper features in the top K suggestions. This metric is pivotal for understanding the efficiency of the prefetching system. A higher R@K value implies that the system is adept at suggesting relevant papers to the users.

6 Results and Discussion

In this section, we present the evaluation results of our prefetching models and compare them with the baseline.Then, we analyze the influence of the number of prefetched candidates to be reranked on the overall recommendation performance.







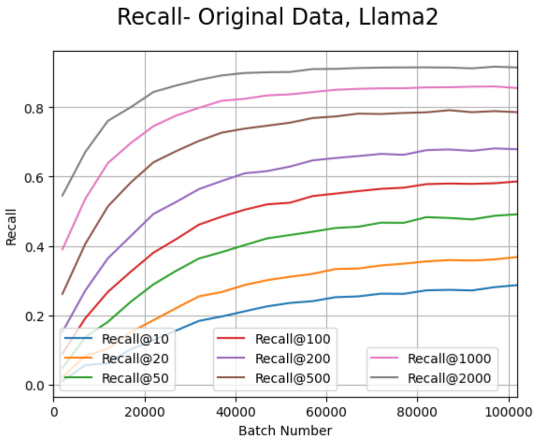
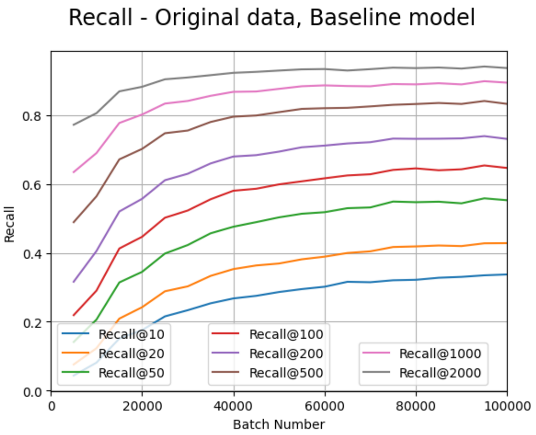


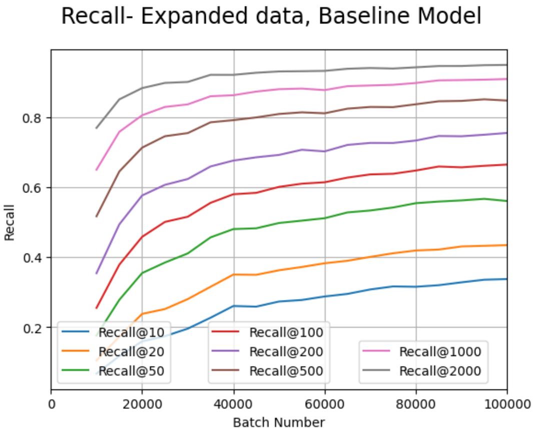
Table 1: Recall - Original data

K	Baseline	BERT	Llama 2	SciBERT
10	0.337	0.300	0.287	0.305
20	0.428	0.391	0.368	0.395
50	0.553	0.514	0.491	0.513
100	0.647	0.606	0.586	0.604
200	0.731	0.696	0.678	0.690
500	0.833	0.804	0.785	0.797
1000	0.895	0.866	0.855	0.862
2000	0.937	0.921	0.914	0.914

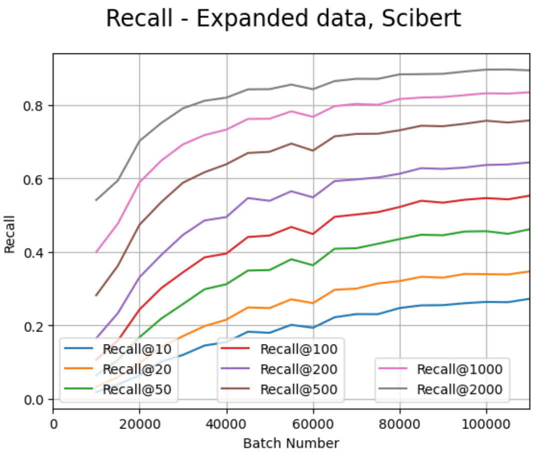
K represents the quantity of candidate papers to be retrieved during the prefetching stage.

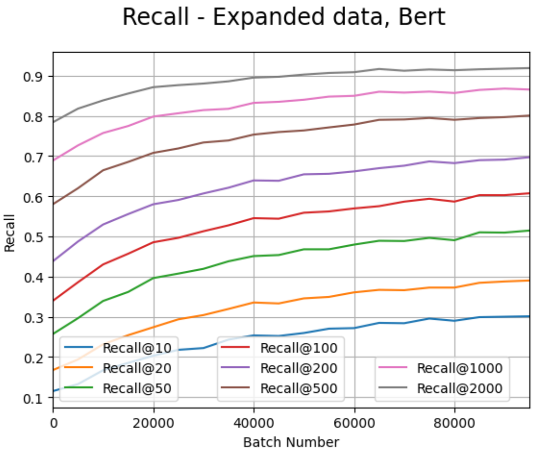
Various prefetching models embeddings were tested. The results indicate that BERT is the most effective embedding for this task. However, please note that even with this improvement, we haven't yet surpassed the results of the baseline article.





Comparison between the baseline model [7] using the original data and the expanded data, resulting in the identification of a slight improvement in the expanded data.





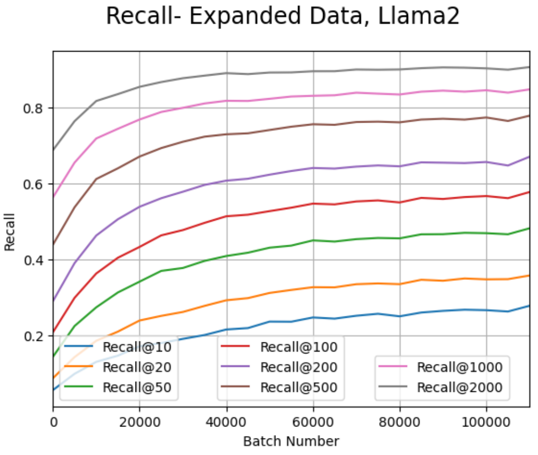


Table 2: Recall- Modified Data

K	Bas - eline	BERT	Llama 2	Sci - BERT
10	**0.337**	0.301	0.277	0.272
20	**0.434**	0.390	0.357	0.346
50	**0.556**	0.514	0.481	0.461
100	**0.664**	0.607	0.577	0.552
200	**0.754**	0.697	0.669	0.643
500	**0.846**	0.800	0.777	0.757
1000	**0.908**	0.865	0.847	0.833
2000	**0.948**	0.918	0.905	0.893

Our expanded dataset exhibits state-of-the-art performance, outperforming the baseline using the original data.

Integration of the previous two experiments to assess the impact of the new prefetching model embeddings on the modified data. The results continue to highlight BERT as a strong performer, although it doesn't outperforms the baseline's [7] superior performance.

7 Future Directions

Looking ahead to an era endowed with greater computational capacity, our team envisions a significant evolution in our methodology—primarily pivoting towards the integration of query expansion techniques. Query expansion, a salient strategy in the domain of information retrieval, augments the initial query with relevant terms and phrases, thereby enhancing the breadth and precision of search results.

Given the inherent complexity and richness of academic papers, leveraging a powerful Language Learning Model (LLM) to facilitate query expansion could be particularly beneficial. Using adapted Reinforcement Learning (RL) technique, The LLM could be trained to discern and introduce pertinent explanatory phrases and contextually relevant keywords from the comprehensive content of the entire paper.

RL offers a dynamic approach to adaptively adjust and optimize the model's behavior based on iterative feedback. With this amalgamation of advanced techniques, we aspire to significantly elevate the efficacy of our citation retrieval mechanism, moving it closer to the ideal prefetching and re-ranking system.

8 Conclusion

In this study, our primary objective was to introduce novel advancements in the domain of Local Citation Recommendation. This was achieved by leveraging diverse encoding techniques and enhancing data quality. By training the baseline model on our improved data-set [7], we attained state-of-the-art results. Our strategy was bifurcated into two phases: the initial phase involved fetching pertinent documents by rapid and efficient encoding methods, while the subsequent phase entailed reranking these documents using the fine-tuned SciBERT model.

References

[1] I. Beltagy, K. Lo, and A. Cohan. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, number 1, pages 3615–3620, Hong Kong, China, Nov 2019. Association for Computational Linguistics.

[2] C. Bhagavatula, S. Feldman, R. Power, and W. Ammar. 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)*, number 2, pages 238–251, New Orleans, Louisiana, Jun 2018. Association for Computational Linguistics.

[3] T. Dai, L. Zhu, Y. Wang, and K.M. Carley. Attentive stacked denoising autoencoder with bi-lstm for personalized context-aware citation recommendation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 28(4):553–568, Jan 2020.

[4] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova. 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)*, number 5, pages 4171–4186, Minneapolis, Minnesota, Jun 2019. Association for Computational Linguistics.

[5] T. Ebesu and Y. Fang. Neural citation network for context-aware citation recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, number 6, pages 1093–1096, New York, NY, USA, 2017. Association for Computing Machinery.

[6] O. Gökçe, J. Prada, N.I. Nikolov, N. Gu, and R.H. Hahnloser. Embedding-based scientific literature discovery in a text editor application. In *Proceedings*

of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, number 10, pages 320–326, Online, Jul 2020. Association for Computational Linguistics.

[7] Nianlong Gu, Yingqiang Gao, and Richard HR Hahnloser. Local citation recommendation with hierarchical-attention text encoder and scibert-based reranking. In *European Conference on Information Retrieval*, pages 274–288. Springer, 2022.

[8] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W.B. Croft, and X. Cheng. A deep look into neural ranking models for information retrieval. *Information Processing Management*, (11):102067, 2019.

[9] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, number 12, pages 421–430, 2010.

[10] G. Herdan. *Type-token mathematics*, volume 4. Mouton, 1960.

[11] W. Huang, S. Kataria, C. Caragea, P. Mitra, C.L. Giles, and L. Rokach. Recommending citations: translating papers into references. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, number 14, pages 1910–1914, 2012.

[12] L. Hunter and K.B. Cohen. Biomedical language processing: what's beyond pubmed? *Molecular cell*, 21(5):589–594, 2006.

[13] C. Jeong, S. Jang, E.L. Park, and S. Choi. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, 124(16):1907–1922, 2020.

[14] T.N. Kipf and M. Welling. 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*, number 18, 2017.

[15] Y. Kobayashi, M. Shimbo, and Y. Matsumoto. Citation recommendation using distributed representation of discourse facets in scientific articles. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, number 19, pages 243–251, New York, NY, USA, 2018. Association for Computing Machinery.

[16] Y. Liu and M. Lapata. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, number 20, pages 5070–5081, Florence, Italy, Jul 2019. Association for Computational Linguistics.

[17] A. Livne, V. Gokuladas, J. Teevan, S.T. Dumais, and E. Adar. Citesight: Supporting contextual citation recommendation using differential search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, number 21, pages 807–816, New York, NY, USA, 2014. Association for Computing Machinery.

[18] Z. Medić and J. Snajder. Improved local citation recommendation based on context enhanced with global information. In *Proceedings of the First Workshop on Scholarly Document Processing*, number 24, pages 97–103, Nov 2020.

[19] V. Nair and G.E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, number 25, 2010.

[20] R.M. Nallapati, A. Ahmed, E.P. Xing, and W.W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, number 26, pages 542–550, 2008.

[21] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142, New Jersey, USA, 2003.

[22] S. Robertson and H. Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Number 30. Now Publishers Inc, 2009.

[23] T. Saier and M. Färber. unarxive: a large scholarly data set with publications' fulltext, annotated in-text citations, and links to metadata. *Scientometrics*, 125(31):3085–3108, Dec 2020.

[24] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number 32, pages 815–823, 2015.

[25] T. Strohman, W.B. Croft, and D. Jensen. Recommending citations for academic papers. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, number 33, pages 705–706, 2007.

[26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, number 34, pages 5998–6008, 2017.

.