Learning to Research: Learning to Ranking the Similar Papers via BERT Fine-Tuning

Jiaxin Ye[1,*], Hong Tian[2,*]

[1]School of Foreign Languages, Dalian Jiaotong University, Dalian, China

[2]School of Software, Dalian Jiaotong University, Dalian, China

[*]Corresponding author e-mail: yejiaxin1020@163.com

[*]Corresponding author e-mail: th@djtu.edu.cn

Abstract. Information retrieval has always been an important research topic in the era of big data. How to accurately retrieve the references needed by scholars from a large number of papers, sort them by relevance, and screen out valuable information to recommend to scholars is an important research demand at present. There has not yet been a model that can capture user attention in academic scenarios based on search results. At the same time, the construction resources of this dataset are limited, and there is a lack of a benchmark dataset that can be used for training search learning models. Therefore, this article constructs a dataset that can be used for academic literature relevance ranking search and a learnable search ranking model. The test results indicate that the model has different advantages in disciplinary fields, confirming that this system can be well applied in academic scenarios and improving the efficiency of scholars in literature search and selection.

Keywords: Learning to Research; Similar Papers; BERT Fine-Tuning.

1. Introduction

Natural language processing（NLP）is an important direction in the field of computer science and artificial intelligence. It studies the theory and methods of effective communication between humans and computers using natural language. The purpose is to enable computers to process or understand natural language for automatic translation, text classification, and sentiment analysis. At the same time, natural language processing is one of the most difficult problems in artificial intelligence. Academic Search has always been a major demand in the academic community, and academic search systems that cater to the needs of professional users such as scholars are particularly important [1,2].

There are various descriptions of the same query and document pair, but if you want to find a more matching answer, the core lies in how to better understand the meaning expressed by the query and document. From the perspective of deep learning, that is, how to represent them as more meaningful vector forms. The existing relevant practice has proved that the word embedding vector can provide more abundant information than the traditional bag-of-words model. However, traditional word vector models such as word2vec [3] and fastest [4] cannot solve the fundamental problem of polysemy, and models based on convolutional neural networks and recurrent neural networks cannot effectively model long-distance contexts. Many transformer-based pre-trained models, represented by BERT[5], perform language modeling by performing multiple pre-trained tasks on a large corpus.

Common retrieval systems are geared towards large-scale retrieval, searching for relevant documents in large datasets. Some use spurious relationship feedback to improve the performance of traditional information retrieval models, and some use correlation matching or semantic matching to improve the retrieval performance of models. However, there is no system that can capture users' attention in specific scenarios yet.

In academic scenarios, if scholars' attention can be captured based on a search result, it can improve the efficiency of scholars in literature search and selection, reduce the unnecessary time for reading complex and repetitive articles, and also support daily updates, allowing scholars to discover more suitable and suitable literature in less time. However, due to the limited resources for constructing datasets in this area, there is a lack of a benchmark dataset that can be used for training search learning

models. Therefore, this article intends to construct a dataset that can be used for academic literature relevance ranking searches.

Based on the above analysis and summary, our contributions are as follows:

We have constructed a new literature retrieval dataset, which has made some contributions to the community in the fields of information retrieval, data mining and natural language processing, and effectively promoted the development of sorting learning.

We proposed a fine-tuned model based on BERT. Through the combination of learning-to-rank (LTR) and a pre-trained language model, we use a sorting algorithm to make the text similarity model reach a certain accuracy.

We proposed a system for academic scenarios. When the final draft is published, our model will be open-source and publicly accessible.

2. Related Work

The article [6] studies whether the pre-trained model still helps tasks in specific fields. It spans four fields and eight classified tasks and finds that the second stage of Domain Adaptation pre-trained can still improve performance. In addition, performing TAPT (task adaptive training) after DAPT can also improve performance. The optimization of TAPT for a single task can damage its migration ability, indicating that the data distribution within a domain may also be different. It also indicates that just conducting DAPT is not enough, and DAPT+TAPT is more effective.

The common sorting learning methods in reference [7] can be divided into three categories. The first category is point based sorting learning methods [8], the second category is pair based sorting learning methods [9], and the third category is list based sorting learning methods [11]. RankNet [12] is a neural network trained using traditional BP and gradient descent algorithm [13,14] for object sorting tasks. LambdaRank [9,10] does not solve the scheduling problem by showing how to define the loss function and then calculate the gradient, but analyzes the physical meaning of the gradient required by the scheduling problem and directly defines the gradient. Kumar et al. [15] proposed an end-to-end training method called ListBERT based on the transformer's RoBERTa model for ranking e-commerce products. Experiments show that compared with other popular list loss functions (such as ListNET and ListMLE), the RoBERTa model using NDCG based proxy loss function fine-tuning (approxNDCG) achieves 13.9% NDCG improvement. Compared with the RoBERTa model based on paired RankNet, the RoBERTa model based on approxNDCG also achieved a 20.6% improvement in NDCG.

In this work, we propose a fine-tune model based on BERT. Through the combination of learning-to-rank(LTR) and pre-trained language model, we use a sorting algorithm to make the text similarity model reach a certain accuracy. We also used the LambdaRank algorithm to fine-tune the BERT based model to compare it with traditional models based on TF-IDF, Doc2Vec, and Siamese Network.

3. Model

In the field of information retrieval[16,17,18], BERT can be used as a pre-trained model for input text, and then the sorting model can be applied to sort search results. Specifically, during the training process, BERT can be used to convert search keywords and text information into vector representations, and then index and vector representations can be input into the sorting model to learn correlation and score using the model. During testing, the input information is converted into vector representations in the same way and sorted based on the scores output by the model to provide the best search results.

In this work, we were inspired by RankNet [12] to use the vectors provided by the pre-trained language model as usual indexes for literature related searches. The search results were then trained and sorted using the learning-to-rank method.

First of all, let's introduce RankNet. This model aims to optimize the inverse logarithm and does not consider the weight of the position. This optimization method is relatively friendly to evaluation indicators such as AUC, but the actual sorting results are not consistent with the actual sorting requirements. In reality, the sorting requirements pay more attention to the correlation of top k. The selection of sorting evaluation indicators such as NDCG is more in line with actual needs. The cross-entropy loss of RankNet, which aims to optimize the inverse logarithm, cannot directly or indirectly optimize indicators such as NDCG. Our work calculates the similarity relationship between vectors and obtains positional weights.

On the basis of RankNet, LambdaRank [9,10] redefined the gradient [19], and we also conducted sorting learning on this basis. LambdaRank is modified on the basis of RankNet. First, the loss function of RankNet is decomposed to obtain the gradient. The decomposition formula is as follows, $w_k$represents the parameters of the neural network model.

$$\frac{\partial C}{\partial w_k} = \frac{\partial C}{\partial s_i}\frac{\partial s_i}{\partial w_k} + \frac{\partial C}{\partial s_j}\frac{\partial s_j}{\partial w_k}$$

$$= \sigma\left(\frac{1}{2}(1 - S_{ij}) + \frac{1}{1 + e^{\sigma(s_i - s_j)}}\right)\left(\frac{\partial s_i}{\partial w_k} - \frac{\partial s_j}{\partial w_k}\right)$$

$$= \lambda_{ij}\left(\frac{\partial s_i}{\partial w_k} - \frac{\partial s_j}{\partial w_k}\right)$$

Lambda can represent the strength of gradients, and Lambda can further simplify it. Assuming that for the paper pairs (i, j) in the training set, the cosine similarity is calculated to obtain the corresponding paper ranking, where paper i is ranked before paper j, S_{ij}=1, Lambda can be simplified as follows.

$$\lambda_{ij} \equiv \frac{\partial C(S_i - S_j)}{\partial s_i} = \sigma\left(\frac{1}{2}(1 - S_{ij}) - \frac{1}{1 + e^{\sigma(s_i - s_j)}}\right)$$

$$\lambda_{ij} = \frac{-\sigma}{1 + e^{\sigma(s_i - s_j)}}$$

Considering that indicators such as NDCG and ERR cannot directly calculate gradients, the gradient Lambda is directly modified to introduce information from evaluation indicators, so that the gradient can approach the performance of evaluation indicators. The approach in the original paper was to exchange the positions of two papers i and j, and then calculate the changes in evaluation indicators |ΔZ|, Take |ΔZ| as the factor of Lambda, and Z is the NDCG evaluation indicator.

$$\lambda_{ij} = \frac{-\sigma}{1 + e^{\sigma(s_i - s_j)}}|\Delta Z|$$

Finally, the loss function is obtained as follows.

$$C = log\left(1 + e^{-\sigma(o_i - o_j)}\right)|\Delta Z|$$

Use this loss function to fine-tune BERT

4. Experiments

4.1 Benchmark

The experimental data of this experiment mainly comes from a total of 2 million articles on COVID-19 and GAKG in the Covidia system [21] and GAKG [20] system, since these two system provide user-friendly API for academic meta-data accessment.

The articles on both websites contain content such as titles and abstracts. The experimental data consists of 10000 different COVID-19 text retrieval data and 10000 different GAKG text retrieval data. The data on the website is encoded using BERT for the input text.

We construct two dataset for geoscience papers and covid-19 related papers and the search terms vary from single keywords to three different keywords. The retrieved results are annotated by determining whether the sentence is related to the search term.

4.2 Baseline

TF-IDF[22]: Represent the text as a word frequency-inverse document frequency matrix, and then use cosine similarity or Euclidean distance to calculate the similarity score.

Doc2Vec[23]: Use PV-DM or PV-DBOW algorithms to represent text as a fixed size vector, and use cosine similarity or Euclidean distance to calculate the similarity score.

Siamese Network[24]: Using a twin network structure to represent two texts as vectors of fixed size, and calculating similarity scores using cosine similarity or Euclidean distance.

When comparing these models, we used the following evaluation metrics to measure their performance:

NDCG (Normalized Discounted Cumulative Gain): Considering the relationship between literature ranking and relevance, it is more suitable for use in ranking tasks than standard accuracy and recall indicators.

Pearson/Spearman correlation coefficient: measures the linear/nonlinear relationship between the predicted values of the model and the true correlation.

RMSE (Root Mean Square Error): Measures the average square error between the predicted values of the model and the true correlation.

The experimental results are as follows

Table 1. experiment on Covidia.

Covidia

	NDCG	Pearson	RMSE
TF-IDF	0.78	0.12	38.21
Doc2Vec	0.87	0.23	28.91
SiameseNetwork	0.86	0.34	22.67
LambdaListBERT	0.91	0.51	19.21

Table 2. experiment on GAKG.

GAKG

	NDCG	Pearson	RMSE
TF-IDF	0.68	0.01	2392.12
Doc2Vec	0.77	0.13	1928.23
SiameseNetwork	0.62	0.02	2191.29
LambdaListBERT	0.81	0.14	1819.75

From the table above, it can be seen that our model outperforms other baselines in the dataset we selected.

5. Conclusion

To sum up, we have successfully constructed a new literature retrieval dataset, which has made some contributions to the community in the fields of information retrieval, data mining and natural language processing, and has effectively promoted the development of sorting learning. This dataset not only contains a large number of subject literature but also provides correlation labels, enabling us to conduct effective model training and performance evaluation. Secondly, we propose a fine-tuned model based on BERT. Through the combination of learning to rank and the pre-trained language model, we use the sorting algorithm to make the text similarity model reach a certain accuracy. In our experiment, we compared our model with traditional models based on TF-IDF, Doc2Vec, and Siamese Network, and found that our model achieved better performance in evaluation indicators

such as NDCG. This indicates that our model has better sorting ability and can more accurately determine the similarity between texts.

References

[1] Gusenbauer, M., & Haddaway, N. R. (2020). Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources. Research synthesis methods, 11(2), 181-217.qualities of Google Scholar, PubMed, and 26 other resources

[2] Orduña-Malea, E., Martín-Martín, A., M. Ayllon, J., & Delgado Lopez-Cozar, E. (2014). The silent fading of an academic search engine: the case of Microsoft Academic Search. Online information review, 38(7), 936-953.

[3] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[4] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. arXiv preprint arXiv:2004.10964.

[7] Li, H. (2014). Learning to rank for information retrieval and natural language processing. Synthesis lectures on human language technologies, 7(3), 1-121.

[8] Li, P., Wu, Q., & Burges, C. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. Advances in neural information processing systems, 20.

[9] Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. Learning, 11(23-581), 81.

[10] Burges, C., Ragno, R., & Le, Q. (2006). Learning to rank with nonsmooth cost functions. Advances in neural information processing systems, 19.

[11] Cao, Z., Qin, T., Liu, T. Y., Tsai, M. F., & Li, H. (2007, June). Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning (pp. 129-136).

[12] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005, August). Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning (pp. 89-96).

[13] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

[14] Taylor, M., Guiver, J., Robertson, S., & Minka, T. (2008, February). Softrank: optimizing non-smooth rank metrics. In Proceedings of the 2008 International Conference on Web Search and Data Mining (pp. 77-86).

[15] Kumar, L., & Sarkar, S. (2022). ListBERT: Learning to Rank E-commerce products with Listwise BERT. arXiv preprint arXiv:2206.15198.

[16] Liu, T. Y. (2009). Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval, 3(3), 225-331.

[17] Xu, J., & Li, H. (2007, July). Adarank: a boosting algorithm for information retrieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 391-398).

[18] Liu, Y., Lu, W., Cheng, S., Shi, D., Wang, S., Cheng, Z., & Yin, D. (2021, August). Pre-trained language model for web-scale retrieval in baidu search. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (pp. 3365-3375).

[19] De Boer, P. T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. Annals of operations research, 134, 19-67.

[20] Deng, C., Jia, Y., Xu, H., Zhang, C., Tang, J., Fu, L., ... & Zhou, C. (2021, October). GAKG: A multimodal geoscience academic knowledge graph. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 4445-4454).

[21] Deng, C., Ding, J., Fu, L., Zhang, W., Wang, X., & Zhou, C. (2023). Covidia: COVID-19 Interdisciplinary Academic Knowledge Graph.

[22] Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, No. 1, pp. 29-48).

[23] Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv:1607.05368.

[24] Chopra, S., Hadsell, R., & LeCun, Y. (2005, June). Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, pp. 539-546). IEEE.