An Effective Approach for Citation Intent Recognition Based on Bert and LightGBM

Weilong Chen*
University of Electronic Science and Technology of China
chenweilong921@gmail.com

Shuaipeng Liu*
Meituan-Dianping Group
liushuaipeng@meituan.com

Wei Bao*
Southeast University
willinseu@gmail.com

Huixing Jiang†
Meituan-Dianping Group
jianghuixing@meituan.com

ABSTRACT

In the development of science and technology, the public scientific theses have played an important role and greatly promoted the development of society. The vast majority scientific progress was announced in the form of papers in past centuries, and impactful contributions were often recognized by the research community with a great number of citations. However, inappropriate citation of papers still occurs from time to time and hinders the progress of human civilization. In this paper, we proposed an effective framework to address the citation intent recognition challenge in ACM WSDM Cup 2020[1]. Our team name is *ferryman* and in our solution, we regarded this problem as the Information Retrieve (IR) task and proposed a framework with two stages of recall and ranking and finally our team won *the 1st place* with a Mean Average Precision @ 3 (MAP@3) score of 0.42583 on the final leaderboard[2].

KEYWORDS

Citation Intent Recognition, Information Retrieve, Nature Language Processing

1 INTRODUCTION

WSDM Cup is a competition-style event co-located with the leading WSDM conference. This paper describes our solution for Citation Intent Recognition, one of WSDM Cup 2020 tasks, and we won the 1st place in the final leaderboard. Science has emerged as a dominant engine of innovation for modern society. Moreover, its rich published traces allow us to understand, predict and guide its advance and utility like never before. Research papers are the dominant media for state-of-art knowledge. Therefore, if we can develop models that understand research papers, we can greatly enhance the ability of computers to understand knowledge.

The competition provided a large paper dataset, which contains roughly 800K papers, along with paragraphs or sentences which describe the research papers. These pieces of description are mainly from paper text which introduces citations. The participants are required to recognize the paper cited in the describe texts. This competition uses Mean Average Precision @3 (MAP@3) as the evaluation metric which is described by the following function:

$$MAP@3 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(3,n))} P(k)) \quad (1)$$

Where |U| is the number of press_id in the test set, P(k) is the precision at cutoff k, n is the number of predicted papers.

After analyzing the challenge, we regard it as an Information Retrieve (IR) task[11], The IR focuses on the problem of finding the most matched Top N documents with a query from a massive number of candidate documents. In this challenge, the description text is the query and the candidate papers are the documents to be retrieved. To handle this challenge, we made a plan with two stages including recall and ranking. In recall stage, several unsupervised methods like Axiomatic F1EXP[5], DFI Similarity[7], Okapi BM25[14] are built to reduce the scope of candidates, then we draw learning to rank models such as BERT[4][10] and lightGBM[6] to ranking the candidate papers which is selected in the recalling stage.

The rest of the paper is organized as follows: Section 2 describes our solution which contains the model details. In Section 3, we show the experiments and results of our model. Finally, we conclude our analysis of the challenge, as well as some additional discussions of the future directions in Section 4.

2 METHODOLOGY

In this section, we introduce our framework for Citation Intent Recognition. Firstly, we introduce the recall strategy. Secondly, we introduce the rank strategy based BERT and lightGBM, Finally We introduce how to integrate the models.An overall framework and processing pipeline of our solution is showed in Figure 1. Our trained models and source code are publicly available on GitHub[3].

2.1 Recall Strategy

In the recall stage, candidate papers and descriptions were represented as a vector using vector space model and bag-of-N-gram model, in practice, the max N is set to 2 owing to the huge computational space. Then we use several similarity measurement to reduce the retrieve scope, including TFIDF, BM25, LM Dirichlet, Axiomatic F3EXP, DFI Similarity, Axiomatic F1EXP, Axiomatic F2EXP, Axiomatic F1LOG, Axiomatic F2LOG, Axiomatic F3LOG, Boolean Similarity, LM Jelinek Mercer Similarity, DFR Similarity, IB Similarity and so on[2][11]. And we apply the structure introduced above

*Both authors contributed equally to this research.
†All the corresponding to Huixing Jiang.
[1]http://www.wsdm-conference.org/2020/wsdm-cup-2020.php
[2]https://biendata.com/competition/wsdm2020/final-leaderboard/
[3]https://github.com/myeclipse/wsdm_cup_2020_solution

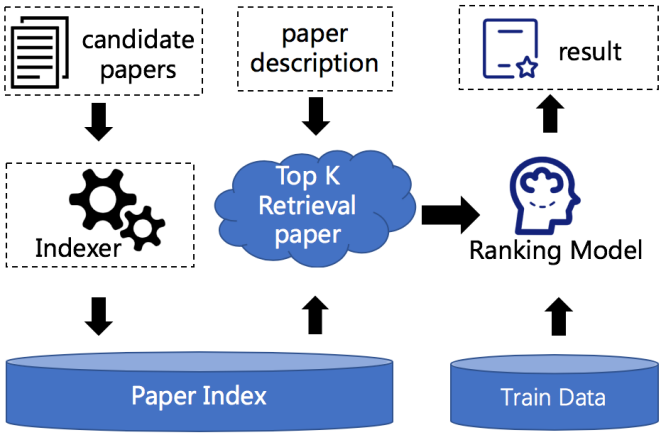

Figure 1: An overall framework and pipeline of our solution for citation intent recognition

on different scales of a paper, such as title, abstract, keywords and full text. In our practice, the F1EXP has the highest recall score and BM25 get the highest MAP score. The recall results is not only used to reduce the retrieve scope but all as a part of features used in the LGB ranking stage.

2.2 BERT Model

The BERT[4][10] model architecture is based on a multilayer bidirectional Transformer[15] As Fig. 2. Instead of the traditional left-to-right language modeling objective, BERT is trained on two tasks: predicting randomly masked tokens and predicting whether two sentences follow each other. BERT model gets a lot of state of the arts in many tasks, and we also use the BERT model in our strategy. There are two types of BERT models following the same architecture as BERT but instead pre-trained on the different scientific texts: SciBERT[1] and BioBERT[9]. Also, we trained the pre-trained model in two ways: The Point-Wise model and the Pair-Wise model.

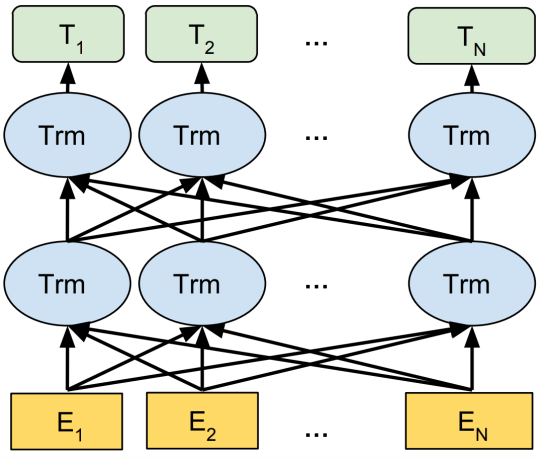

Figure 2: Bidirectional transformer architectures of BERT

2.2.1 Data Preprocessing. The better preprocessing of the input can get better performance. Firstly, we removed the excess white-space and some stop words, and we did some word segmentation and did part-of-speech tagging. Secondly, we normalized the word form for the different tags of the sentence and lowercased all letters. We compared the input without preprocessing and the input with preprocessing, finding that the input with preprocessing is better than another one.

2.2.2 Bert with Point-Wise. We trained the BERT with Point-Wise way which means we defined the task as the binary classification. We preprocessed the two sentences (the description sentence and the paper-described sentence). We joined them in one sentence with [SEP] token and put them into the BERT model. We trained the token of the sentence with binary cross-entropy loss to dig the difference between description sentence and paper-described sentence As Figure 3. The probability can measure how well the two sentences match. However, too much negative samples can destroy the performance of the BERT model and the Point-Wise way didn't take into account the internal dependencies between the documents corresponding to the same query. On the one hand, the samples in the input space are not independently identically distribution. On the other hand, the structure between these samples was not fully utilized. When different queries correspond to different numbers of documents, the overall loss will be dominated by the query group with a large number of documents. Each group of queries should be equivalent. We need to have another way to get better performance of the model. We tried the Pair-Wise model.

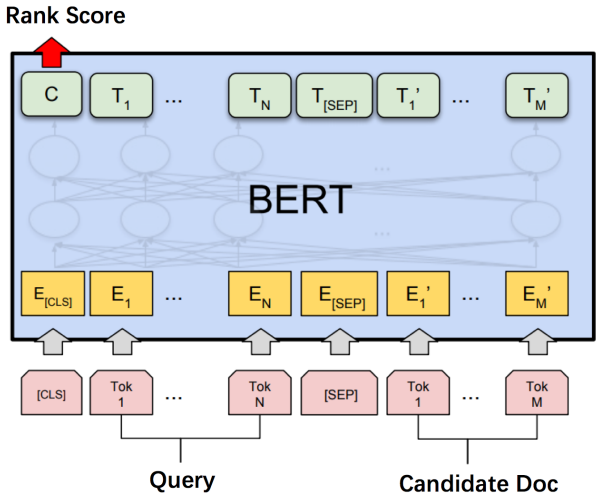

Figure 3: Ranking with BERT

2.2.3 Bert with Pair-Wise. Learning2Rank applies machine learning technology to the ranking problem and trains the ranking model. Usually, the discriminant supervised machine learning algorithm is applied. Learning2Rank task seeks ranking results and does not require precise scoring, as long as there is a relative scoring. Learning2Rank framework has the following characteristics:

- The samples in the input space are two feature vectors (corresponding to the same query) composed of two documents (and corresponding query).
- The samples in the output space are pairwise preference.
- The samples in the space are two-variable functions and the loss function evaluates the difference between the predicted preference and the true preference of the document pair.

We did the same preprocessing to the input sentence as the way described in the above. We used the margin ranking loss as our loss function and trained several triplet samples with the same description text and different paper-described sentences. It not only helped to get a better ranking of similarity but also compared the differences between each description text. We got a higher score than the BERT model with Point-Wise.

2.3 Lightgbm Model

In order to increase the diversity of the model, in addition to Bert, we choose LightGBM for modeling, and for simplicity, it is called lgb here. lgb model is a gradient boosting framework that uses tree based learning algorithms. LightGBM builds the tree in a leaf-wise way, as shown in Figure 4, which makes the model converge faster.LightGBM is not sensitive to outliers and can achieve high accuracy, which is widely used in industry. And in this work, compared with Bert, the effect of LightGBM is better, the LightGBM single-model can reach 0.413 in the leaderboard. Total number of features is 1684, this contains of semantic features, statistical features and so on, which will be explained later.

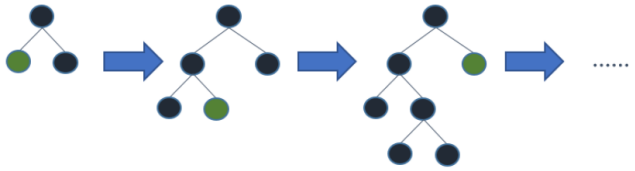

Figure 4: LightGBM's leaf growth strategy

In this work, the training method of LightGBM is lambdarank(pairwise strategy), which is about 0.5% higher than the traditional binary classification model(pointwise strategy). The following will be carried out from two aspects of feature engineering and model construction.

2.3.1 feature engineering. Our feature engineering mainly consists of the following 3 aspects:

- *Semantic feature.* Semantic features include various pre-trained word vector models such as fasttext[3], glove[13], word2vec[12], doc2vec[8] etc. And we retrain them to calculate the similarity between description and abstract. Specifically, we represent the vector of a sentence as the average of the word vectors of each word in it. Then we use the cosine distance formula and the Manhattan distance formula to measure the correlation between the two sentences, and the correlation value is used as our semantic feature.
- *Statistical features and word frequency features.* In this section, we use various word frequency-based methods to capture similarities, such as bm25, tfidf, f1exp and various length and proportion features. Among these word frequency features, we find that the similarity obtained through the bm25 method is very important. At the same time, compared with the semantic features, the word frequency features bring greater benefits to the model as a whole. We believe this is due to the large number of specialized terms in the corpus.
- *Rank features.* In order to make our model easier to "know" the essential purpose of ranking, we sort the various similarity values according to description_id (or paper_id), and divide the ranking value by the number of description_id (or paper_id) to get the relative ranking ratio. This part of can bring a 3% boosting. In detail, suppose we have m correlation features. Then through our grouping and sorting operation, since we can group according to description_id or paper_id, we can get another 2m new sorting features, and divide by the corresponding number in the group, we can also get another 2m new sorting scale feature.

2.3.2 Modeling Methodology. Since the same description can recall multiple paper abstracts, from the perspective of a classification problem, this is an imbalance of positive and negative samples, so the number of samples cannot be too large. However, in the composition of the training set, we found that the positive sample coverage ratio of the recall samples is also very important, so we chose a higher number of recall samples. At the same time, in the training set, because some descriptions cannot recall the positive samples through our recall strategy, we artificially added the positive samples to the training set in order to ensure the coverage of the positive samples. Through the above data preprocessing steps, the amount of training data for lgb model is about 5 million.

Learning to Rank is one of the most commonly used algorithms to implement ranking through machine learning. It mainly includes three types of single document method (pointwise), document pair method (pairwise) and document list (listwise). The pointwise single-document method means it will judge the relevance of each document to this query, and converting the documents ranking problem into a classification (such as related, irrelevant) or a regression problem. However, the pointwise method does not learn other document as features when modeling, so it cannot consider the order relationship between different documents. The purpose of rank learning is mainly to sort the documents in the search results according to the magnitude of relevance, so pointwise is bound to have some defects.

Aiming at the problem of pointwise, the pairwise document method does not care about the specific value of the correlation between a document and a query, but converts the ranking problem into any two different documents related to the relative order of the current query. In order to be relevant and irrelevant, the two categories are recorded as +1, 0, and then transformed into classification problems. Listwise treats all related documents corresponding to a query as a single training sample.

In total, Our Lgb model is trained using a 5-fold cross-validation method. The training target is lambdarank, and the offline verification indicators are MAP @ 3 and MAP @ 5. The model score can reach 0.413.

2.4 Ensemble Methodology

In the model ensemble stage, we adopted a simple and efficient way and get 1.2% boosting. We group the model prediction results of LightGBM and BERT by description id, and then add the the ranking values with weighting operation, the weights of which are 6 and 4, respectively. The details are shown in Figure 5.

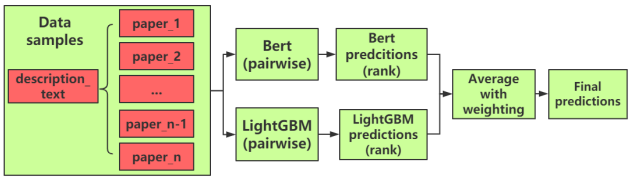

Figure 5: Ensemble strategy based on rank blending with weighting operation

3 EXPERIMENT

3.1 Experimental Settings

In this experiment, our training set has a total of about 63,000 paper description documents, and its number on the test set is about 34,000. At the same time, our candidate paper dataset has a total of about 840,000 papers.For each piece of description, we need to choose 3 best-matching papers in candidate paper dataset.

Table 1: Online map@3 score with different models

Model	Online MAP@3 LB score
Bert(pointwise)	0.397
Bert(pairwise)	0.402
LightGBM(pointwise)	0.405
LightGBM(pairwise)	0.413
Ensemble	0.425

3.2 Model Comparison

Here we compare the performance of our method with different settings. The results are shown in Table 1.From the table, we can see that no matter in Bert or LightGBM, the result of pairwise training method is better than pointwise. At the same time, the LightGBM model based on detailed feature engineering is very effective. Our best single mode is LightGBM trained using pairwise methods, which is reflected in the algorithm settings as lambdarank.

At the same time, our highest score is the ensemble model of the Bert model and LightGBM model. We noticed that the improvement based on ensemble between LightGBM models is very limited, but the Bert model and LightGBM model can bring a huge improvement of 1.2%, which we believe is due to the huge difference between the two models.

4 CONCLUSION

In this paper, we propose a method based on Bert and LightGBM for recognition of paper citations, in which both Bert and LightGBM are trained using pairwise methods. At the same time, we won the first place in the Citation Intent Recognition competition (WSDM Cup 2020 track1).

ACKNOWLEDGEMENTS

We thank everyone associated with organizing and sponsoring the WSDM Cup 2020. Dataset was provided by Microsoft Research. Challenge was sponsored and managed by the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020). Competition platform was hosted by Biendata. we are very grateful to WSDM Cup Chairs Kyumin Lee and Neil Shah for their great efforts during the challenge.

REFERENCES

[1] Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676* (2019).
[2] Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. 2012. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*. 17.
[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[5] Hui Fang and ChengXiang Zhai. 2005. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 480–487.
[6] Guolin Ke, Qi Meng, Thomas William Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tieyan Liu. 2017. LightGBM: a highly efficient gradient boosting decision tree. (2017), 3149–3157.
[7] İlker Kocabaş, Bekir Taner Dinçer, and Bahar Karaoğlan. 2014. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information retrieval* 17, 2 (2014), 153–176.
[8] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. arXiv:cs.CL/1405.4053
[9] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746* (2019).
[10] Shuaipeng Liu, Shuo Liu, and Lei Ren. 2019. Trust or Suspect? An Empirical Ensemble Framework for Fake News Classification. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining, Melbourne, Australia*. 11–15.
[11] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.
[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:cs.CL/1301.3781
[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162
[14] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.
[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.