Abstract
Recent research on the semantic retrieval task shows that the pretrained language models like BERT have impressive re-ranking performance [4]. In the re-ranking process, the fine-tuned language model is feed with (query, document) pairs and the whole time complexity is directly proportional to both the query size and the recall set size. In this paper, we describe a simple yet effective strategy of early stopping strategy based on the confidence score. In our experiment, such strategy can avoid up to 30% unnecessary inference computation cost without sacrificing much ranking precision. The codes and documents are available at https://github.com/chengsyuan/WSDM-Adhoc-Document-Retrieval. Our team dlutycx ranked first on the unleak track.

Keywords. Passage Re-ranking, Semantic Retrieval, WSDM Cup 2020

1 Introduction
Modern search engines retrieve web documents mainly by matching keywords in documents with those in search queries. Though they are fast enough, lexical matching can be inaccurate due to the academic sentence rewriting (some researchers may rewrite the sentence to reduce the rate of duplicate checking). In this situation, a concept can be an expression in different vocabularies, and hence the lexical similarity can be fairly low.

To achieve better search results, the semantic information of the query and the document are considered in the re-ranking process. The DSSM model [1] is first proposed to use the multi-layer neural network to map the high-dimensional text features into low-dimensional dense features. This method significantly outperforms the traditional lexical ranking model, e.g., TF-IDF and BM25. To further capture the semantic information, convolutional neural network architecture is imported to represent the query and the document [5]. Their model demonstrates strong performance gains on the twitter re-ranking task. And more recently, the transformer-based models, such as BERT, have achieved more impressive results on the passage re-ranking task MS-MACRO [4].

Though these models show impressive re-ranking performance gains, in the production or time-limited competition, it is only practical to re-rank several documents at top k position of the recall set. In this paper, we propose a simple yet effective early stopping strategy for the re-ranking process.

2 Pipeline
Our solution for the ad-hoc document retrieval consists of three main stages:

- Cleaning: the documents with missing data are removed and the texts unrelated to the task are also deleted.
- Recalling: a recall set to a given question is retrieved from the whole candidate document database by an unsupervised manner, such as BM25 or document embedding similarity.
- Re-ranking: each of these documents is scored and re-ranked by a more computationally-intensive method.

2.1 Cleaning
In the cleaning step, we simply remove the missing data. Then we clean the text which is not directly related to the topic. Concretely speaking, we remove each of the sentences that is not a citation (**##**).

2.2 Recalling
In the recall step, we use the Okapi BM25 [2] to measure the lexical similarity between query and document. The formula is as follows:
\[ RSV_d = \sum_{t \in q} \text{log} \left[ \frac{N}{dt} \right] \cdot \frac{(k_1 + 1) \cdot tf_{td}}{k_1 ((1 - b) + b \times (L_d/L_{ave})) + tf_{td}} \]

where \( k_1 \) suppresses the long document. After several experiments on the validation set, we set \( k_1 = 2 \) and \( b = 0.75 \).

2.3 Re-ranking

In the re-ranking step, we use the pretrained BioBERT [3] to get the similarity score [4]. The model details are illustrated in Figure 1.

Relevant or Not

![Diagram](image)

Figure 1. We truncate the query to have at most 64 tokens. We also truncate the passage text such that the concatenation of query, passage, and separator tokens have the maximum length of 512 tokens. Then we use the [CLS] vector as input to a single layer neural network to obtain the probability of the passage being relevant.

Then, binary cross-entropy loss is adopted to fine-tune the BioBERT:

\[ L = - \sum_{j \in J_{pos}} \log (s_j) - \sum_{j \in J_{neg}} \log (1 - s_j) \]

where \( J_{pos} \) is the set of indexes of the relevant passages and \( J_{neg} \) is the set of indexes of non-relevant passages in top-20 documents retrieved with BM25. To balance the pos-neg rate, we over-sample the positive documents 19x.

After fine-tuning the BioBERT, we use this model as a fixed scorer \( f(q,d) \) during the re-ranking inference. In the following algorithm, we describe the regular re-ranking strategy which is widely used:

As shown in Algorithm 1, the regular re-ranking strategy is to simply iterate over every document in the recall set. As we can observe in Figure 2, the truth documents are not uniformly distributed. They aggregate on the top position. In order to solve such a problem, we design an early stopping strategy. As shown in Algorithm 2, when the re-ranker (fine-tuned BERT model) shows high confidence, we can assume this document as the most relative document.

Algorithm 1 Regular Re-ranking Strategy

Input:
- A scorer \( f(q,d) \) to measure the similarity;
- A query \( q \);
- The recall set \( D(q) \) of the query;

Output:
- The re-ranked documents list \( R(q) \)

1: score = emptyList();
2: for each \( d \in D(q) \) do
3: score.append\( f(q,d) \);
4: end for
5: Sort and Return \( R(q) \);

Figure 2. The heatmap shows the top 200 recall set and each pixel represents a document of a given query in a row. The white pixels indicate relevant.

As shown in Figure 3, the distribution of the maximum scores is not the same as Figure 2. If the Algorithm 2 is adopted in the re-ranking process, we may mis-retrieve the irrelevant document as a positive document if the score of the false positive-document is higher than the threshold.

To mitigate such a problem, we propose an adaptive early stopping re-ranking strategy as shown in Algorithm 3. We believe that the experience-based batch size \( b \) can reduce the false-positive documents and achieve the MAP@3 gain (mean average precision at 3, which is the metric of the leaderboard).

3 Experiments

3.1 Dataset

The competition provides a large paper dataset, which contains roughly 800K papers, along with paragraphs or sentences which describe the research papers. These pieces of
Figure 3. The heatmap shows the relevant score of each (query, document) pair scored by the previously fine-tuned BioBERT model.

Algorithm 2 Early Stopping Re-ranking Strategy
Input:
A scorer $f(q, d)$ to measure the similarity;
A query $q$;
The recall set $D(q)$ of the query;
A threshold $t$ set by experience;
Output:
The re-ranked documents list $R(q)$

1: $\text{score} = \text{emptyList}();$
2: for each $d \in D(q)$ do
3: $\text{score}.append(f(q, d));$
4: if $f(q, d) > t$ then
5: Break;
6: end if
7: end for
8: Sort and Return $R(q);$

description are mainly from paper text which introduces citations.

For example:
Description: An efficient implementation based on BERT [1] and graph neural network (GNN) [2] is introduced.

Related Papers:

Our task is to retrieve the related paper from the candidate set (800K papers), given the description.

3.2 Validation Of Our Proposed Method
To validate our proposed method, we conduct several experiments on a tiny validation set with 100 samples. As shown in Table 1, the regular re-ranking strategy can be considered as the performance upper-bound with the highest time cost. In contrast to the regular one, our proposed early stopping strategy shows the least time cost while showing the highest performance loss. A possible explanation is the existence of the false-positive document. A higher threshold $t$ set by experience or adopting the adaptive re-ranking strategy would be helpful. Finally, experiments show that the eclectic adaptive re-ranking strategy saves about 40% re-ranking time and sacrifices the least performance.

3.3 Test Set
We select several commits on the test leaderboard. As shown in Table 2, the regular strategy shows the least MAP@3. With the increasing $k$, MAP@3 increases.

Table 1. Comparison of three algorithms, tested on 100 samples. The empirical threshold $t$ is set to 3.0.

<table>
<thead>
<tr>
<th>Method</th>
<th>$K$</th>
<th>MAP@3</th>
<th>Time Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular Strategy</td>
<td>5</td>
<td>0.4217</td>
<td>1x</td>
</tr>
<tr>
<td>Regular Strategy</td>
<td>10</td>
<td>0.4633</td>
<td>2x</td>
</tr>
<tr>
<td>Regular Strategy</td>
<td>50</td>
<td>0.5350</td>
<td>10x</td>
</tr>
<tr>
<td>Regular Strategy</td>
<td>100</td>
<td>0.5217</td>
<td>20x</td>
</tr>
<tr>
<td>Early Stopping</td>
<td>5</td>
<td>0.4217</td>
<td>0.5x</td>
</tr>
<tr>
<td>Early Stopping</td>
<td>10</td>
<td>0.4533</td>
<td>1x</td>
</tr>
<tr>
<td>Early Stopping</td>
<td>50</td>
<td>0.5200</td>
<td>5x</td>
</tr>
<tr>
<td>Early Stopping</td>
<td>100</td>
<td>0.5067</td>
<td>10x</td>
</tr>
<tr>
<td>Adaptive Early Stopping</td>
<td>5</td>
<td>0.4214</td>
<td>0.6x</td>
</tr>
<tr>
<td>Adaptive Early Stopping</td>
<td>10</td>
<td>0.4667</td>
<td>1.4x</td>
</tr>
<tr>
<td>Adaptive Early Stopping</td>
<td>50</td>
<td>0.5350</td>
<td>6.6x</td>
</tr>
<tr>
<td>Adaptive Early Stopping</td>
<td>100</td>
<td>0.5217</td>
<td>13.0x</td>
</tr>
</tbody>
</table>
4 Conclusion

In this competition, we implemented a simple but effective system to solve WSDM - DiggSci 2020 competition, and our end-to-end pipeline is clean and high extensible without the help of feature engineering. Our solution finally ranked 2nd in the validation set and 4th in the test set, and without data leak, our solution ranked 1st in the test set.

Our solution can be summarized as follows.

1. We performed data cleaning on the dataset according to self-designed saliency-based rules, and removed the redundancy data.
2. Before reranking, we used the bm25 metric. For faster calculating, we adopted the cupy to accelerate the calculation. Compared with traditional algorithm, the matrix multiplication with cupy can be done in 15 minutes using a single GPU card.
3. We used the fine-tuned BioBERT model to score every (query, document) pair and designed a novel early stopping strategy for re-ranking based on the confidence score to avoid up to 40% unnecessary inference time cost of the BERT.

5 Future Work

Though the adaptive early stopping strategy is effective, there are still two hyper-parameters to set by human experience. It is worthy to develop a method to make our proposed method fully automated.

6 Acknowledgments

We really appreciate the help from Yanning Shen, who provided a 8-GPU server for 4 days. Also specially thanks to Haibo Wang for his effort on the proofreading.

References