Relevance Matching vs Semantic Matching

Jay Chien
b07902021@ntu.edu.tw
DRMM dataset / data cleaning / BM25

Jing-Neng Hsu
b08902022@ntu.edu.tw
Word2Vec model / report

Po-Heng Chen
b07902114@ntu.edu.tw
BERT dataset, training, and experiments / data cleaning

Yi-Min Lin
b07902016@ntu.edu.tw
DRMM training, experiments

1 Introduction

In recent years, deep learning models help breaking the limitation of traditional methods in various fields. Since powerful NLP models have already been proposed, we wonder that whether we can apply the models for ad-hoc retrieval tasks. In our project, we reproduce two models. The first one is Deep Relevance Matching Model (DRMM[3]), which is a relevance matching based approach. And the second one is BERT-based retrieval model, which is an application of semantic matching on ad-hoc retrieval. Finally we evaluate the performance of the two methods, and discuss the pros and cons of each matching scenario. Whether the combination of the two methods can perform better is remained future work.

2 Related Work

There have been neural methods for relevance matching since about 2017, including soft n-gram based model PACRR[5], counting based model like KNRM[6] and DRMM[3]. All of them claims to beat traditional methods like BM25.

As for semantic matching methods, [9] is the first application of BERT[2] on ad-hoc retrieval task. And there are many further improvement for this, like BERT-MaxP[1] and PARADE[7], which has the best performance among BERT-based models. There's method that combines relevance matching and semantic matching, taking representation from fine-tuned BERT as the input of neural based models, like CEDR[8], which has comparable performance with PARADE.

3 Methodology

3.1 Bert for retrieval

Basic idea. If a document and a query are relevant, it is highly possible that the semantic information in them are related to a certain extent. Therefore, we want to use deep learning technique to extract the semantics from text. Bert is the state of the art when it comes to contextual analysis. It is pre-trained on very large scale data, and the pre-trained task include finding relationship between sentences and predicting words in sentence. After knowing Bert's ability, we would like take use of it to construct our retrieval model by using semantic matching between query and document.

3.1.1 Data Preprocess.

Training data. First, we split the whole tokenized document into multiple passages with overlapping. In addition, owing to that one passage can't be too short to keep enough information, we set the window size of each passage to 230 tokens at most, and the splitting stride to 115 tokens.
Second, for each document and query pair in Trec-robust04 dataset, we concatenate the tokenized query with each passage of the document, and add with special tokens([CLS] at begin, [SEP] between query and pasaage). By this process, we convert query and document pairs to training data which we can use as Bert input.

Target. A passage is labeled positive if it belongs to a relative document to the query, negative otherwise. We make a assumption that if a document is relative to a query, all the passages in the document are also relative with that query, and vice versa. Although it is such a strong assumption, it would be reasonable if we choose suitable window size for each passage. Besides, when we use model to predict if document is relevant with a query, we would use pooling technique to get the final relative score from each passages, and it can also make the result not affected much by this assumption.

Model and training. Input the training data we make in previous steps to Bert, and turn the task into sentence pair classification problem. Simply apply MLP after CLS token hidden state and fine-tune entire model so that the MLP output would fit the target.

Predict. After getting prediction label for each passage and query pair, we have to convert passages relevance score back to document relevance score. We define three pooling method to do this.

- Max passage score pooling (MaxP) : Use max score among all passages in the document as document relevance score.
- Mean passage score pooling (MeanP) : Use mean score of all passages in the document as document relevance score.
- Ensemble pooling : Linear combination between MaxP score and MeanP score.

3.2 DRMM

Basic idea. If a query matches a document, the terms in the query and the ones in the document should be similar. DRMM is a relevance matching model that focuses on the interaction of each query term and document term. Besides, DRMM computes the importance of each query term, and important terms contribute more to the final relevance score.

3.2.1 Data Preprocess.

Training data. First, we train a word-to-vector model on a stemmed, lower-cased, punctuation-removed corpus. Then, we convert all the terms in queries' title and documents into embedding indexes, and compute the IDF (Inverse Document Frequency) for each index.

Target. In [3], Jiafeng Guo el at. mentioned three problem for semantic matching: **1. not using exact matching, 2. no query term importance, 3. the matching is in global scope**. The target of the model is to consider exact matching, query term importance, and matching in local scopes.

3.2.2 Model. DRMM consists of 4 stages: local interaction, histogram matching, feed forward, and score aggregation.

Local interaction. We compute the cosine similarity score of each term in query and in document using term embedding, and pass the interaction matrix to histogram matching stage. The matrix is in size of *(query length × document length)*, where each element [i, j] records the similarity score between the i-th query term and the j-th document term.

Histogram matching. This stage discretize the similarity score into 30 intervals between [-1, 1]. For an query term q, we count the number of scores in matrix[q] appearing in an interval, and apply logarithm over the count value, resulting in an vector of size 30, as the extent of similarity between q and the document.

Feed forward. After histogram matching stage, we feed the histogram vectors of the query into a fully connective network, and output hidden score vector z for the terms.

Score aggregation. In this stage, we input the original query signals into one linear layer, with softmax as activation function, and output g as the weight for z in the previous stage. After getting g, we can compute the relevance score of the query and the document by $g \cdot z$. There are 2 kinds of query signals: **1. Term vector**, the embedding of the terms. **2. IDF**, the IDF of the terms.

3.2.3 Training. We use hinge loss the maximize the score of the positive documents and minimize the score of the negative document.

$$L(q, d^{+}, d^{-}) = \max(0, 1 - s(q, d^{+}) + s(q, \grave{d}))$$

4 Experiments

4.1 MAP comparison between different method

Table 1. MAP comparison between different method

Method	MAP
Okapi	0.231
DRMM$_{IDF}$	0.180
DRMM$_{TV}$	0.198
BERT$_{MaxP}$	0.280
BERT$_{MeanP}$	0.284
BERT$_{Ensemble}$	**0.298**

The table above shown the average MAP score of 50 testing queries between different method on reranking top 2000 relevant documents ranked by okapi/BM25. The MAP score of BERT outruns other methods.

4.2 Case Observation

Although the MAP score of BERT is relatively high compare to other two methods, we discover that for some queries, BERT actually yields a worse performance, the table below displays those cases(the value of (Okapi, DRMM) is the number of queries that Okapi has better performance than DRMM):

Table 2. Number of queries that one method outperforms another

	Okapi	DRMM	BERT
Okapi	0	28	16
DRMM	22	0	12
BERT	34	38	0

4.2.1 For the query that DRMM performs well.

Title *Organic soil enhancement*

Description *Identify documents that discuss the use of organic fertilizers (composted sludge, ash, vegetable waste, microorganisms, etc.) as soil enhancers.*

Local Interaction. We visualize the local interactions between the query terms and the document terms as in Figure 1.

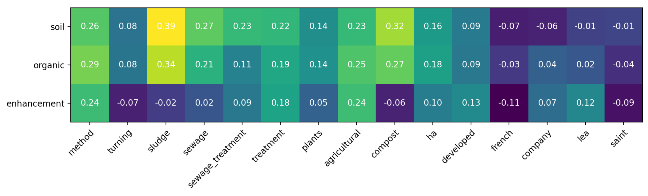


Figure 1. Local Interaction of the DRMM

We notice that even there's not any exact match, the query term "soil" has more interaction with the document terms such as "sludge", "compost" and "agricultural", which are the terms that matches the query description. And we think it's benefited from the fact that word-embedding is trained on the corpus, and it may capture the similarity of terms that appear at specific topics.

Term Importance. There are two choices for the query signal as the input of gating network. And the choice of "Term Vector" and "IDF" may affect the result a lot.

Table 3. Term importance of DRMM$_{TV}$, which results in MAP score of 0.289

Term	Term Importance
soil	1.047
organic	1.040
enhancement	-2.850

Table 4. Term importance of DRMM$_{IDF}$, which results in MAP score of 0.010

Term	Term Importance
enhancement	6.076
organic	5.178
soil	4.679

Using the same scores output by feed forward network as input, DRMM$_{TV}$ outperforms DRMM$_{IDF}$ a lot. And the only difference between them is the choice of query signal for gating network. Which implies that **term importance largely affects the retrieval results**.

In this case, the term "soil" should be considered more important. Otherwise, like the results returned by DRMM$_{IDF}$, contains documents about organisms, chemical fertilizers, and which are not relevant.

4.2.2 For the query that DRMM performs terribly.

Title *Educational Standards*
Description *There has long been a call for standards in U.S. education, these calls frequently citing the superiority of foreign school systems. Are there many countries outside the U.S. which have standards for pre-teen students? If so, which are those countries and what standards have been set?*

Table 5. Term importance for query "Educational Standards"

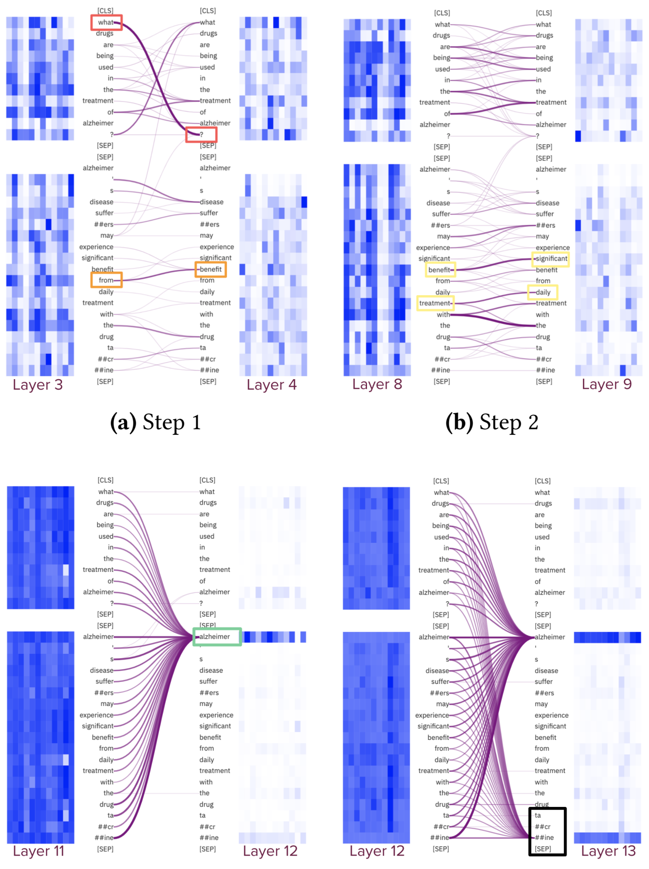
Term	Term Importance
teen	6.151
students	2.313
school	2.280
frequently	0.606
standards	0.253

In this case, DRMM uses query's description as input and get MAP score of 0.019. And we found that the term importance is strongly biased toward "teen", "students" and "school", which results in documents about "children-care", "school" but not relevant to "educational standards".

The first reason is that using description may lead to more noise, and the second reason is that **DRMM consider the importance for each term independently**, which makes term importance not adaptive. In contrast, contextualized embedding may capture more semantic information here.

4.2.3 How Bert retrieve? Although we know that Bert is state-of-the-art in contextual analysis and it does performs well in our experiment, limited discussion on score doesn't demonstrate how Bert model extract meaning of sequence text and how it find the relationship between queries and documents. Therefore, we use exBERT[4] to look deep into attentions among each layers (**Figure 2**, Query description : What drugs are being used in the treatment of Alzheimer?; Retrieved document : Alzheimer's disease sufferers may experience significant benefit from daily treatment with the drug Tacrine.), and then we classify what Bert actually do into 3 step according to the behavior of the attention between layers.

- Step 1 : Getting the structure of input (Layer 1~4)
  - resolving coreference
  - getting relationship between preposition and other words (**Figure 2-a**, red box)
  - finding problem's interrogative word (**Figure 2-a**, orange box)
- Step 2 : Analyzing the inputs meaning (Layer 5~9)
  - getting meaning relationship between input words (**Figure 2-b**, yellow box)
  - Similar to what bag of word do
- Step 3 : Finding relevance, both word level matching and semantic level matching, between query and document (Layer 10~13)
  - Getting the important keyword in document (**Figure 2-c**, green box)
  - Finding the semantic answer to the query (**Figure 2-d**, Black box)



(a) Step 1 **(b)** Step 2

(c) Step 3 **(d)** Step 3

Figure 2. Attention between layers

4.2.4 Bert pooling type discussion. Table 6 is our experiment about different precision of three Bert pooling type. We can see that Bert$_{meanP}$ has better performance than Bert$_{maxP}$. The reason is that even if the document's meaning is far from query, there is still a chance that there is a passage whose semantic match query's. In this case, Bert$_{maxP}$ is easy to make mistake, but Bert$_{meanP}$ would not be affected because it averages all passage scores. However, when the mean semanctic matching scores of many documents are close, Bert$_{maxP}$ can help a lot. Using the best passage score to rerank those documents is such a reasonable method, and this is also why Bert$_{Ensemble}$ can get higher precision.

Table 6. Precision of each Bert pooling type

	Bert$_{maxP}$	Bert$_{meanP}$	BERT$_{Ensemble}$
P@10	0.468	0.456	**0.490**
P@20	0.445	0.436	**0.460**

5 Conclusions

The general performance is BERT > BM25-Okapi > DRMM in our experiment. However, there are still some queries where DRMM outperforms, which means the problems in semantic matching Jiafeng Guo et al.[3] proposed do exist. In section **4.2.1**, focusing on the term-interaction rather than the sentence-semantic helps a lot to retrieve the right document.

BERT performs well on general case, it proves the assumption in section **3.1**. Our experiments show why and how BERT can extract the semantic meaning. However, for the queries like section **4.2.1**, BERT still need to deal with the problems when using semantic matching. Especially when user use several incoherence words as query. It is hard for Bert to perform well when query is not a sentence with complete meaning because Bert rely on contextual information to extract semantic.

In our experiments, semantic matching dominates relevance matching, but some of the queries still needs the technique of relevance matching. To improve both models, combining them together may be a good attempt.

References

[1] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. *CoRR* abs/1905.09217 (2019). arXiv:1905.09217 http://arxiv.org/abs/1905.09217

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[3] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Oct 2016). https://doi.org/10.1145/2983323.2983769

[4] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2019. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276* (2019).

[5] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1049–1058. https://doi.org/10.18653/v1/D17-1110

[6] Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. *CoRR* abs/1810.12936 (2018). arXiv:1810.12936 http://arxiv.org/abs/1810.12936

[7] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage Representation Aggregation for Document Reranking. *CoRR* abs/2008.09093 (2020). arXiv:2008.09093 https://arxiv.org/abs/2008.09093

[8] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. *CoRR* abs/1904.07094 (2019). arXiv:1904.07094 http://arxiv.org/abs/1904.07094

[9] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. arXiv:1903.10972 [cs.IR]