Final Report for Artificial Neural Network

Yingzhuo Qian
Department of Computer Science and Technology
qyz17@tsinghua.edu.cn

Yibo Shen
Department of Computer Science and Technology
EricSam413@outlook.com

Abstract

In order to better solve the IR problem, we adopted BERT initialization and GAT attention into Entity-Duet Neural Ranking Model, which represents queries and documents by their words and entity annotations. We used BERT to initialize single words and GAT attention mechanism to initialize entities. Then we used KNRM to get the ranking. As for proper words, we used an exact match local model as implementation. Our experiment is based on MRR@10 dataset.

1 Introduction

Information retrieval(IR) is an important subfield in NLP, characterizing by the need of retrieving information from tons of data with limitied input. As the amount of data continues to grow in a rapid speed, obtaining useful information has become an ever-accelerating need. Our work focus on the task of passage ranking, which aims to search for passages most relevant to a given query from numerous candidates. In real practice, this task is crucial to search engines, a daily-used system which is bothered by the inaccuracy of passage retrieval results. Therefore, improvement on the accuracy of passage retrieval shall prove significant to search engines, which is the incentive of our work.

During this task, we've met a great deal of challenges and was, unfortunately, unable to solve all of them. The first challenge encountered was the vastness of data. As the dataset we used has over 270G of training data, it must be seperated into different parts. Also, since our model involves not only the word but also the entity representation of data, it is essential to send them into the ranking model well organized. To solve this, we took rather long time to design a DataLoader class and preprocess the training data, segmenting the data to ensure that it can be loaded into the main memory.

Another difficulty is the memory limit of GPU on the server. The GPUs on the server provided by MEGVII has only 10G memory each, which is insufficient to train our model. To overcome this, we had to sacrifice the size of embedding dimension and our entity graph, which conceivably affected our model's performance.

The last, and the most thorny trouble resulted from the lack of time. Due to the terribly overwhelming course loads in Week 10-15, we started coding rather late, which proves to be a huge mistake. As mentioned above, data processing took considerable amount of time, expecially the entity recognition of passages and queries. Recognizing 2.9G worth of passages and queries using tagme with 64 threads shall take approximately 80 hours, and actually took longer due to network failures. After the model is all set up, GPU memory problem kept the training unstarted. After tackling the problem, during which modification of model and entity graph and attempt of distributed learning were make, we were finally able to begin at the start of week 17. However, it appears that training with bert and GAT requires significant amount of time. We had only finished 1/20 of the whole 270G train set in

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Wednesday, and the performance of our model at that time was not convincing. Some hyperparameters were tuned but the result was still unsatisfactory. For the pressure of final exams, we had to give up and admit our failure.

2 Related works

With the rapid development of deep learning techniques, neural IR has shown remarkable advantages over traditional IR techniques like **BM25**. It leverages word embedding, instead of discrete bag-of-words, to present a distributed representation of text, and is hence far better in text feature capturing.(Xiong et al., 2016). Neural IR approaches in passage ranking can be categorized into two types: representation-based models which measure the relevance between representation of queries and passages;(Shen et al., 2014). Interaction-based models which directly measure the term-level relevance of queries and passages(Guo et al., 2016). It has been found that using kernal pooling with gaussian kernals in interaction-based models can significantly enhance the capability to extract soft match features in multiple levels(Xiong et al., 2016; Dai et al., 2018). Studies also show that combination of local model, a simple model capturing exact term matches, and distributed model outperforms either of them working solely(Mitra et al., 2017)

Compared with traditional *word2vec* or *GLOVE* embedding, Bidirectional Encoder Representations from Transformers(**BERT**) representation is far more powerful in extracting text features(Devlin et al., 2018). The fast development of large scale knowledge graph has further improved the performance of search system. The embedding of entities can be learned through modeling their relations as translation operating on embedding space(Bordes et al., 2013). The recognition of Wikipedia entities can be done with Tagme, a system providing online entity recognition(Ferragina and Scaiella., 2010). In incorporating entities in interaction-based models, the model learns copius latent semantic information which is not provided in the word embedding of queries and passages.(Liu et al., 2018) Since entities are presented in the form of a graph where the relations between entities are modeled as edges, GNN can be incorporated to gain a augmented representation of entities. Graph attention network(GAT) is an architecture operated on graphs using attention mechanism(Velickovic et al., 2017), which can be used to augment entites's semantic using their neighbours.

3 Method

As shown in Figure 1, our model is comprised of two parts: local model and distributed model. Our distributed model is depicted in Figure 2.

We first present our BERT-EDRM model, which serves as the distributed model. In this model, we consider query $q = v_1, v_2, ...v_m$ and passage $d = v_1, v_2, ..., v_n$. We obtain word-level representation by using pretrained BERT model and get $Q_{m\times d}, D_{n\times d}$, where d is the dimension of BERT hidden layer.

For each query and passage, we use *tagme* to obtain its Wikipedia entities. We use OpenKE-Wikipedia embedding as the inputs of our GAT network. Then we use a one-layer-GAT network to get the final embedding, which is augmented by the semanic information of its neighbors and itself with certain weights.

$$\overrightarrow{h_i^{l+1}} = ||\sigma(\sum_{j_i} \alpha_{ij}^k W^k \overrightarrow{h_j^l}) \tag{1}$$

Where the h^l is initial embedding, h^{l+1} is the augmented entity embedding, $\alpha_{ij}^k$is the kth weight in attention mechanism.

Then, we use word and entity representation of query and passage to generate interaction matrix.

$$M^{ij} = cos(\overrightarrow{v_{h_q}^i}, \overrightarrow{g_{h_d}^j}) \tag{2}$$

At last, we use KNRM to generate rating. For each interaction matrix M, we get

$$K(M_i) = K_1(M_i), K_2(M_i), ..., K_k(M_i) \tag{3}$$

and

$$\phi(M) = \sigma_{i=1}^n log(bn(K(M_i))) \tag{4}$$

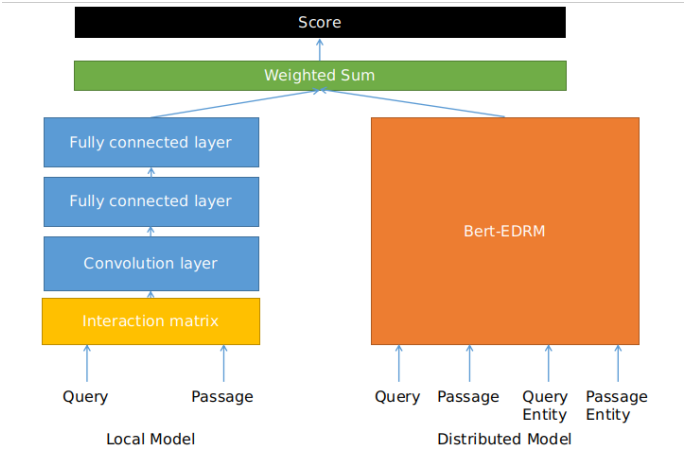

Figure 1: general model

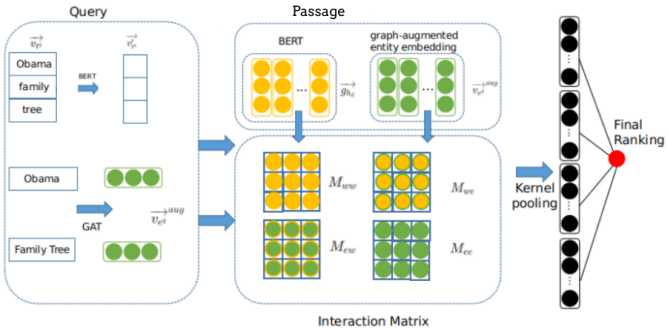

Figure 2: BERT-EDRM

Different from its original implementation, we add a batch normalization layer to avoid gradient vanishing or gradient exploding. Then, we concaternated every $\phi(M)$ and get

$$\Phi(M) = \phi(M_{q_w,d_w}) \oplus \phi(M_{q_w,d_e}) \oplus \phi(M_{q_e,d_w}) \oplus \phi(M_{q_e,d_e}) \tag{5}$$

and loss function

$$f(q,d) = tanh(\omega_r^T \Phi(M) + b_r) \tag{6}$$

As for local model, we encoded the query and passage into $m \times 1$ matrix $Q = [q_1, q_2, ..., q_{n_q}]$ and $D = [d_1, d_2, ..., d_{n_d}]$ by one-hot encoding. Then, we generated a $n_d \times n_Q$ matrix $X = D^T Q$ which engulf every possible exact match. Then, this matrix is multiplied by a convolution layer, a tanh layer and two fully connected layers to output the final ranking $f_l(q)$. Because local model is prone to highly related queries and passages, we only give them a subtle weight. The last ranking function is

$$g(q,d) = (\alpha + \beta) f_l(q,d) + (1 - \alpha) f(q,d) \tag{7}$$

where α, β are trainable weights.

For each query, we generated one positive passage and one negative passage to output the final rating function

$$l = \sigma_q \sigma_{d^+,d^- \in D_q^+,-} max(0, 1 - g(q,d^-), +g(q,d^+)) \tag{8}$$

Given positive and negative passages as input, the model can be trained end-to-end using pairwise hinge loss.

4 Experiment

4.1 Dataset

Our dataset is MS-MARCO, which contains real Web documents and user queries from search engine *bing*. This set is comprised by train, dev and eval with a rate of 8:1:1. For dev and eval parts, there

are approximately 6800 queries, each with approximately 1000 passages generated by method BM25. As for entities, we used OpenKV-Wikipedia dataset with 50 dimentions. Our goal is to find out top 10 related ranking passafes. Our dataset is generated by method MR@10.

4.2 Baseline

Our baseline model is *acl2018: Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval.* This model uses CNN to construct unigram, bigram and trigram representation of query and passage words, and incorporates entity relation, type and description embedding to obtain an enriched-entity representation. Then, it leverages KNRM to get the final ranking score.

4.3 Parameters

Our parameters are listed below:

Table 1: pretrained bert

attention_probs_dropout_prob	0.1
hidden_act	"gelu"
hidden_dropout_prob	0.1
hidden_size	768
initializer_range	0.02
intermediate_size	3072
max_position_embeddings	512
num_attention_heads	12
num_hidden_layers	12
type_vocab_size	2
vocab_size	28996

Table 2: GAT,KNRM

head_num	1
layer_num	1
entity_embedding_dimension	50
kernal_num	5

Table 3: train parameter

learning_rate	5e-5
batch_size	8
dropout	0.6
optimizier	Adam
passage_len	80
query_len	15
passage_ent_size	2
query_ent_size	12
gradient_accumulate_step	4
eval_step	1000

As for local model, we used 100 filters in the first convolutional layer. The parameters in the fully connected layers are determined by the length of passage and query.

4.4 Training process and Result

Unfortunately, as mentioned before, we didn't managed to generate final results. Our failure is partly due to the restricted time and space. Apart from the former reasons, the small capacity of Server GPU also impeded our process. In order to fit our entity into two 10G-GPUs, we resized the former model from 20 million entities to 3 million with formal id, then to 640 thousand in use and at last,

210 thousand which was mentioned over 5 times. This was a sinuous process and took up a long period of time.

However, we managed to run a small fraction of the entire dataset, but the result was terrible. We attributed our failure to the distribution of our dataset. Because it was generate from real passages, the distribution of training set may have large distance with another evaluation set when considering only a small fraction. If given more time, the results may be better since the whole training set and the whole evaluation set shouldn't have too large distance with each other.

Due to the mere time, we didn't manage to evaluate on official dev set. We used another part of training set as dev and got the following results in the 4000 and 8000'th step:

Table 4: results

step	4000	8000
accuracy	0.4980	0.5320

The evaluate set is comprised of 1000 untrained training set members.

The original training log is attached.

5 Conclusion

In this task, we deeply felt the difficulty of solving real problems with nerual networks, as the size of data is remarkably large the distribution is extremely hard to learn. We conclude from our failure that experiments must start as early as possible, and that the construction of network is at most half of the work.

5.1 Devision of work

Yingzhuo Qian: proposed model； chose dataset; preprocessesd MS Marco data and completed entity recognition; implemented data loader for model; implemented KNRM and integrated all 3 models; composed train and eval scripts and conducted experiments; collaborated in writing project proposal, milestone report and final report.

Yibo Shen: implemented local model and GAT; preprocessed OpenKE entity embeddings into graph structure; resized entity graph for the need of cutting memory expenses; collaborated in writing project proposal, milestone report and final report.

References

Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017, August). End-to-end neural ad-hoc ranking with kernel pooling. In Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval (pp. 55-64). ACM.

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014, November). A latent semantic model with convolutional-pooling structure for information retrieval. In Proceedings of the 23rd ACM international conference on conference on information and knowledge management (pp. 101-110). ACM.

Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016). A Deep Relevance Matching Model for Ad-hoc Retrieval. conference on information and knowledge management,, 55-64.

Dai, Z., Xiong, C., Callan, J., & Liu, Z. (2018). Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. web search and data mining.

Mitra, B., Diaz, F., & Craswell, N. (2017). Learning to Match using Local and Distributed Representations of Text for Web Search. the web conference,, 1291-1299.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv: Computation and Language,.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Advances in neural information processing systems (pp. 2787-2795).

Ferragina, P., & Scaiella, U. (2010, October). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 1625-1628). ACM.

Liu, Z., Xiong, C., Sun, M., & Liu, Z. (2018). Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. meeting of the association for computational linguistics.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph Attention Networks. arXiv: Machine Learning,.