Information Retrieval Models for Passage Re-ranking Part 2

19138867
Department of Computer Science, University College London
London, UK

ABSTRACT

This project is to address the fundamental problem of short text retrieval and relevancy re-ranking with given query text in information retrieval system. And the main goal of this project is to develop and improve the retrieval model with using learning to rank algorithms. In this project, information retrieval models are developed by using Logistic Regression, LambdaMart ranking algorithm and Convolutional Neural Network. And evaluation metrics, mean average precision (MAP) and normalized distributed cumulative gain (nDCG) are developed to measure the learning to rank retrieval models' performance.

KEYWORDS

Information Retrieval, Short Text Matching, Re-ranking

1 INTRODUCTION

Information retrieval is an important infrastructure component among diverse kinds of applications, ranging from web search engine, library system to recommendation system. With the booming of data resource, information retrieval becomes much important than ever before, which it is supposed to retrieve as much as possible documents/passages which are relevant to a given query. In order to increase the effectiveness and efficiency of retrieval process, preprocessing the short text pair and utilizing a pre-trained word embedding model are first step for developing learning to rank algorithms. During the preprocessing, the raw texts are split by non-word characters, tokenlized and stemming for unifying the words which might have same meaning. And, word embedding is proved to be effective in representing words' semantic meaning and statistical property related to others words. Because of the limited size of dataset, the pre-trained words embedding, Glove, is used as word representation for machine learning based information retrieval models implementation. In this paper, three learning to rank retrieval model are developed for short text relevancy ranking, which are Logistic Regression, LambdaMart in xgboost and convolutional neural network. Two evaluation metrics are developed to measure the models' performance. The developed information retrieval model are tested under a query-passages candidate reranking dataset from the URL [1].

This paper will be formulated as following. Section 2 is the problem formulation of text re-ranking tasks, which will introduce the key component in learning algorithms and data processing. Section 3 introduces the learning to rank methodology and evaluation metrics used in the experiment. In section 4, the experimental set up and the results obtained from the information retrieval models will be discussed. In the last section, it is the conclusion of this project.

2 PROBLEM FORMULATION

This section explains how to formulate the information retrieval problem by using machine learning knowledge and how to tackle the retrieval problem in practice. The basic idea of learning to rank model is to learn a function

$$f(w, g(q_i, D_i))- > \text{Ranking Score}$$

where w is the learned weights of models, q_i is i-th query from the dataset, D_i is the candidate passages, $D_i = \{q_1, ..., q_N\}_{i=1}^{N}$ to be ranked of q_i and $g(q_i, D_i)$ is the representation for the query-passage pairs in the dataset.

2.1 Learning to Rank

There are mainly three approaches for learning to rank models, which are pointwise, pairwise and listwise. Pointwise methods is the simplest one among the learning approaches, which can be regarded as training a classifier to determine if a passage is relevant to a given query or not. And models are optimized directly from the given labels of the candidate passage. Pairwise methods is to train the learning model to keep the partial order of the relevancy for two query-passage pairs. If the relevancy score of query-passage i is higher than the relevancy score of another query-passage j, the learning model will produce scores where

$$f(w, g(q_i, D_i)) > f(w, g(q_j, D_j))$$

And the last approaches, listwise method, will unify a specific query with all its candidate passages as the input of the learning model. Then, the model will produce a learned ranking on the candidate passages and optimized with respect to ground truth absolute ranking of the candidate passages' labels. In this project, the Logistic Regression retrieval model and Convolutional Neural Network (CNN) are trained using pointwise approach. And the LambdaMart model is a classical listwise learning to rank approach.

2.2 Words Representation

Learning a good representation of words is essential to improve the performance of learning to rank model. And Glove embedding is used as the basis of learning a representation vector of query-passage pairs. Glove embedding is one of the most popular pre-trained embedding model used in various machine learning based natural language processing task as well as information retrieval tasks, which capture the semantic meaning and syntatic regularities of words. [4] Passage retrieval task values the occurrence of key words in the candidate text list, and semantic relation between query and candidate passage is also an important indicator for relevancy assessment. Therefore, Glove embedding is good choice to exploit the representation of query-passage pairs.

Since Logistic Regression and LambdaMart in xgboosting are two traditional machine learning technique, the feature engineering is necessary for the representation of query-passage pairs. Average

[1] https://drive.google.com/file/d/1npkPA-BdiGELHfBrUOcpqumjbQTspg9p/view

embedding weighted by terms' tf*idf score is used to represent a specific query or passage. The reason is that the weighted average embedding can eliminate the noises in the final average embedding introduced by the common words and stop words. Since key words of query might occur too many times in the candidate passage list, hence low tf*idf score, the log scale term frequency of non stopwords overlap between query and passage are therefore included in the query-passage pair representation. For the CNN method, the raw words embedding of terms in query and passage are stacked together and fed into the model directly.

2.3 Imbalanced Dataset

The query-passage relevancy dataset is highly imbalanced, with 0.1% of query-passage pairs are relevant. This section will introduces the efforts spent on addressing data imbalanced problem. He et.al [2] gives a comprehensive overview of dealing with imbalanced dataset. Inspired from that overview, the oversampling of minority class in the dataset and weighted loss function are used in this project. In logistic regression and convolutional neural network method, oversampling is to sample the relevant query-passage pair repeatedly without replacement from the original dataset and keep the relevant class as a fixed ratio in the generated data for model training. And the loss of each sample in training data will be scaled by the inverse ratio of its ground truth label class in the generated training data.

3 METHODOLOGY

This section will briefly explain the methodology used to develop and evaluate the information retrieval system.

3.1 Evaluation Metrics

Two evaluation metrics, average precision and normalized distributed cumulative gain (NDCG), are developed to measure the performance of retrieval model. Precision is a measure for unranked lists, which is obtained by

$$Precision = \frac{TF}{TF + FP}$$

where TF is true positive number of retrieved results and FP is false positive number of retrieved results. And average precision is the average of precision for relevant document at the ranked list. When an irrelevant document is retrieved, the precision at corresponding position is 0. NDCG is discounted cumulative gain (DCG)against to the DCG of ground truth ranking of candidate passage. DCG at rank k is calculated by

$$DCG@k = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)}$$

where rel_i is the relevancy of the document retrieved at rank i-th position.

3.2 Logistic Regression

Logistic regression is a statistical model to assess the probability of an item belong to a certain class. And the probability is produced by

$$f(W, X_i) = \frac{1}{1 + e^{-\sum_{i=j}^{k} w_j * x_{ij}}}$$

where W is trainable parameter of linear combination and X_i is the vector representation of the query-passage pair i. In this project, cosine similarity of query's and passage's tf*idf weighted average term embedding as well as the log scale term frequency of words overlap between query and passage are used as the vector representation of a given query-passage pairs. Since relevancy assessment in pointwise approach can be formulated as a binary classification problem, the logistic regression retrieval model is optimized by cross-entropy loss

$$loss = \frac{1}{m} \sum_{i=1}^{m} y_i log(f(W, X_i)) + (1 - y_i) log(1 - f(W, X_i))$$

where y_i is label of i-th query-passage pair. Training query-passages pairs are randomly samples from the training dataset regardless of the differentiation in query.

3.3 LambdaMart and Xgboosting

LambdaMart is a learning to rank algorithm combining boosting tree model and LambdaRank algorithm. [1] Boosting tree model is an ensemble method which train and combine several decisions tree for regression/classification problem. And LambdaRank algorithm is to train a ranking model based on the gradient of its current nDCG, where a large change of nDCG sores by changing the position of training item in the output ranking list is defined as large gradient. Therefore, LambdaMart is basically a listwise learning to rank methods which take a query and all its candidate passages as a single training instance. LambdaMart ranking model is from xgboost library and hyperparameter tuning is performed. The hyperparameter selection is essential for the performance of boosting tree algorithm. And grid search is used for this tasks, which feed all the combination of candidate parameters to a cross validation model selection model.

3.4 Neural Network

Neural network have proven its success in various field including computer vision, natural language processing as well as information retrieval. And, unlike recurrent neural work has excellent performance on semantic analysis of text and analyze the time series data, convolution neural network performs well on pattern recognition. Thus, CNN is good choice for this shor text matching and ranking problem. For this passage re-ranking task, Severyn et. al [5] proposed a CNN model for this specific task. This network firstly learns a representation for query and passages separately through convolution filter filter and a similarity matrix of query and passage firstly, then concatenates the learned representation together to assess the similarity of query-passage pair. The network architecture is in Figure 1.

The input of the network have size of $R^{|q_s|x|d|}$ for query and $R^{|p_s|x|d|}$ for passage. $|q_s|$ and $|p_s|$ are the length of query and passage and $|d|$ is the size of words embedding. Wide convolution is used in this architecture with adding zero padding $(m-1)x|d|$ before and after the terms embedding of text, and Kalchbrenner

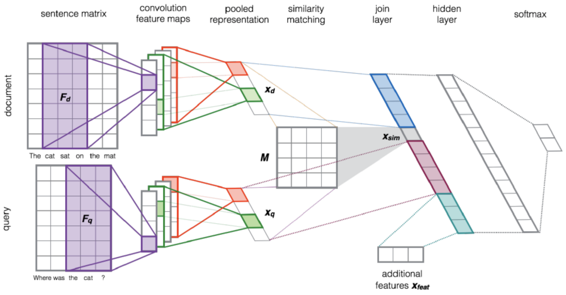

Figure 1: Network Architecture of Convolutional Network from [5]

et.al [3] shows that the wide convolution ensure all the sentence and its margin are reachable for the trainable weights, which is similar to linguistic feature detector m-gram and m is the size of 1d convolutional filter. The 1d convolution goes throught the temporal direction of the text sequence.

The similarity scores of query and passage is produced by a similarity matrix M, where $score_{sim} = A_q^T M A_p$. A_q and A_p are pooled representation at the Figure 1 and M is a size $|d|x|d|$ trainable matrix. Also, this network allows additional feature input which can give an implicit signal to model for learning and greatly improve the performance of ranking task which will be discussed more in details in the following section.

4 EXPERIMENT

The developed models are tested at the dataset got from URL [2] with using the developed metrics, average precision and nDCG. For text preproceesing, the raw text will be changed to its lower cases counterpart and tokenlized. The representation of query/passage is the tf*idf weighted average words embedding of non-stop words in the raw text. For Logistic Regression and LambdaMart model, the cosine similarity of the weighted average embedding between query and passage, and log scale term frequency of the overlap words in the query-passage pairs are used as the representation of query-passage pairs. The mean average precision (MAP) and nDCG scores of the top 100 retrieval results (nDCG@100) in all the retrieval task from the validation dataset are reported as the performance of the developed model.

In logistic regression, the model training is optimized by SGD with a learning rate 1e-4. Although the logistic regression model can achieve similar performance as BM25, it is highly unstable and depends on the weights initialization very much. The reason is that the representation of relevant query-passage pair have lots of overlap with the representation of irrelevant query-passage pair, which is visualized at Figure 2. The cosine similarity and log scale term frequency can only slightly differentiate the relevant item to those irrelevant. Two runs results of logistic regression retrieval model are presented. In training, the relevant samples and irrelevant samples take up the same 50% of the oversampled training dataset, which can prevent the classifier from giving preference to any specific class.

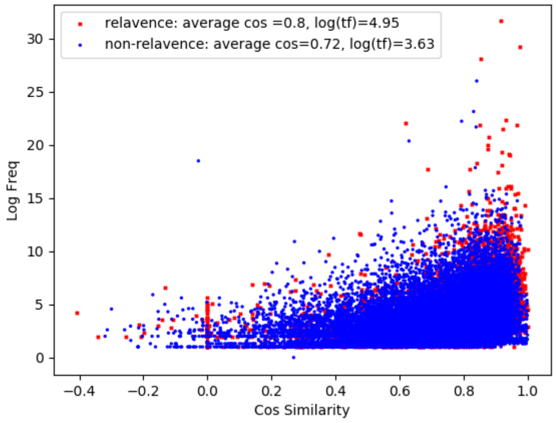

Figure 2: Visualization of relevant and irrelevant query-passage pairs

In LambdaMart model, the parameter is selected by grid search with 5-fold cross validation. Sine the grid search requires a high computational costs, the training instances is sub-sampled to only keep 10% of its total candidates passages. Training and validation instances are also sampled from the current training and testing splits of cross validation. The best parameter setting obtained from the grid search cross validation is with using 'ndcg' as objective function, eta=0.5, gamma=1.0, min_child_weight=5 and max_depth = 6. The detailed explanation of each parameter can be seen from xgboost official documentation at the URL [3].

In neural network model, query and passage are tokenlized to keep a fixed size 10 and 40 respectively by eliminating all the stop words and non-word characters. The kernel size of the 1d convolution is 5, stride size is 1 and the number of feature map is 100. Since the CNN model training need to go through thousands of training iteration, there are only 10% of relevant samples in the oversampled training dataset in order to delay occurrence of overfitting. In order to compensate the class imbalance, the weighted cross entropy loss is used, where the loss of each training items are multiplied by the inverse of its ground truth label's class ratio in the training dataset. The CNN network is optimized using Adam optimizer with learning rate 1e-4. The performance of model is evaluated by using additional feature and without using additional features. The additional feature are cosine similarity of tf*idf weighted average embedding of query and passage as well as the log sclaed term frequency of overlap words in query-passage pair.

The performances reuslts, MAP and nDCG@100, of the developed model on the validation dataset is at Table 1. Based on the performance results, Logistic regression can achieve comparable results as the BM25 model and LambdaMart model outperform the

[2] https://drive.google.com/file/d/1npkPA-BdiGELHfBrUOcpqumjbQTspg9p/view

[3] https://github.com/dmlc/xgboost/blob/master/doc/parameter.rstlearning-task-parameters

Model Name	MAP	Mean nDCG@100
BM25	0.09	0.179
Logistic Regression Run 1	0.146	0.228
Logistic Regression Run 2	0.112	0.187
LambdaMart	0.138	0.228
CNN without additional feature	0.019	0.055
CNN with additional feature	0.073	0.143

Table 1: Performance of models measured by MAP and mean nDCG@100

BM25 model and is the best model in this project. The CNN model, however, leaves behind the BM25 model. The fined tuning of model or better representation learning should be further developed in order to improve the performance. Since the additional features can greatly improved the performance, a better feature engineering might also needed for the performance improvement.

5 CONCLUSION

This project is to develop several learning to rank algorithms, including logistic regression, LambdaMart algorithm and convolutional neural network, for passage retrieval and short text ranking. And the performance of models are measureed by mean average precision (MAP) and the mean nDCG score for the ranking of top 100 results (nDCG@100). All the implemented models can achieve better or at least comparable performance compared with BM25 model. The dataset is highly imbalanced, thus oversampling of minority class and weighted training loss are utilized for addressing the imbalanced nature of dataset. Since learning a word embedding from the relatively small dataset we have is infeasible, pre-trained word embedding Glove is used as the basis for learning the representation of each query-passage pair in the dataset. The cosine similarity of the tf*idf weighted average embedding and the log scale term frequency of overlap words are used as the representation of query-passage pairs for logistic regression and LambdaMart model. And the representation is also used as the additional feature feeding into CNN model to improve the performance.

ACKNOWLEDGMENTS

This report is for the Assignment 2 of COMP0084 Information Retrieval and Data Mining.

REFERENCES

[1] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report MSR-TR-2010-82. https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/

[2] H. He and E. A. Garcia. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (2009), 1263–1284.

[3] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 655–665. https://doi.org/10.3115/v1/P14-1062

[4] 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

[5] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 373–382. https://doi.org/10.1145/2766462.2767738