COMP0084 Coursework 1

Anonymous ACL submission

Abstract

None

1 Task 1: Evaluating Retrieval Quality

1.1 Text Processing

We preprocess passages/queries in two files: *validation_data.tsv* and *train_data.tsv* by following steps:

1. remove url.
2. lower characters.
3. remove non alpha characters.
4. tokenization by Python package NLTK.

1.2 BM25 Model

BM25 Model with parameters implemented in the Coursework 1 is used to retrieval the top passages for each query. Here we retrieval top 3, 10 and 100 passages from 1000 passages. The complete result of BM25 model is in *bm25_raw_top1000.tsv*, while we retrieval top 3 passages for each query in *bm25_ordered_top3.tsv*, top 10 passages in *bm25_ordered_top10.tsv* and top 100 passages in *bm25_ordered_top100.tsv*.

1.3 Metrics

1.3.1 Average Precision (AP)

AP is the average precision of relevant passages for a query.

$$AP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{N} \tag{1}$$

N: number of relevant passages for the query.
k: rank of the passage.

1.3.2 Mean AP

Mean AP is the average of AP over all queries.

$$mAP = \frac{\sum_{q=1}^{N_q} AP_q}{N_q} \tag{2}$$

q: the q_{th} query

1.3.3 Discounter Cumulative Gain (DCG)

DCG is the total gain accumulated at a particular rank p.

$$DCG_q = \sum_{i=1}^{q} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{3}$$

p: particular ranking length
i: the ith passage
rel_i: relevance score

1.3.4 Normalized DCG (NDCG)

Normalizes DCG against the best possible DCG result (the perfect ranking) for a query.

$$NDCG_q = \frac{DCG_q}{optDCG_q} \tag{4}$$

1.3.5 Mean NDCG

Mean NDCG is the average of NDCG over all queries.

$$mNDCG = \frac{\sum_{q=1}^{N_q} NDCG_q}{N_q} \tag{5}$$

N_q: number of queries
The results of the metrics of BM25 model are in Table 1.

Table 1: Metrics of BM25 model

BM25	Cutoff	mAP	mNDCG
Top 3	3	0.1830	0.2007
Top 10	10	0.2250	0.2870
Top 100	100	0.2367	0.3548

2 Task 2: Logistic Regression

2.1 Subsample

The train data set is unbalanced (1% positive, 99% negative). To balance the data set and reduce the training time, we subsample the negative data set to 20 passages per query and keep all positive passages. For these queries with less than 20 negative passages, we keep all negative passages. The subsampled data set has 95874 passages for 1000 queries.

2.2 Word Embedding

We choose Word2Vec to generate word embedding for each term in the vocabulary. The word embedding is a 100-dimensional vector. We use the pre-trained word embedding and set the window size to 5. The word embedding is trained on train and validation data set separately.

2.3 Logistic Regression

Logistic function:

$$\sigma_{\mathbf{w}}\left(\mathbf{x}_i\right)=\left(1+e^{-\mathbf{w}^{\top}\mathbf{x}_i}\right)^{-1} \tag{6}$$

with weight w. The loss function is cross-entropy loss function $\mathcal{J}(\mathbf{w})$:

$$-\frac{1}{n}\sum_{i=1}^{n}\left[y_i\ln\left(\sigma_w\left(x_i\right)\right)+\left(1-y_i\right)\ln\left(1-\sigma_w\left(x_i\right)\right)\right] \tag{7}$$

The gradient of the loss function is:

$$\frac{\partial\mathcal{J}(\mathbf{w})}{\partial\mathbf{w}_j}=-\frac{1}{n}\sum_{i=1}^{n}\left[x_{i,j}(y_i-\sigma_w\left(x_i\right)\right] \tag{8}$$

We use batch stochastic gradient descent to train the model and stop our training when the validation loss does not decrease for 3 epochs. The word embedding of a query-passage pair given by two vectors with 100 dimensions. We concatenate the two vectors and add a bias term to the model and get a 201-dimensional vector.

We initialize the weight vector with 0, set the batch size to 5000, the tolerance to 1e-8, and the maximum number of epochs to 100. To assess the effect of learning rate on our training, we vary the learning rate from 0.01 to 0.0001 and plot the training loss curve in Fig 1. A larger learning rate results in faster convergence and smaller train loss in the initial epoch. However, all the learning rates converge to the same loss value (0.2) after 100 epochs, except for 0.0001, which still shows no sign of convergence. To confirm convergence, we train the model with 0.0001 for 1000 epochs and present the results in Fig 2. The loss converges to 0.2 within 1000 epochs.

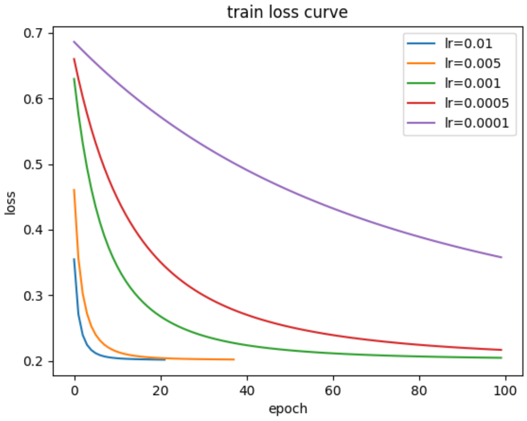

Figure 1: Training loss curve

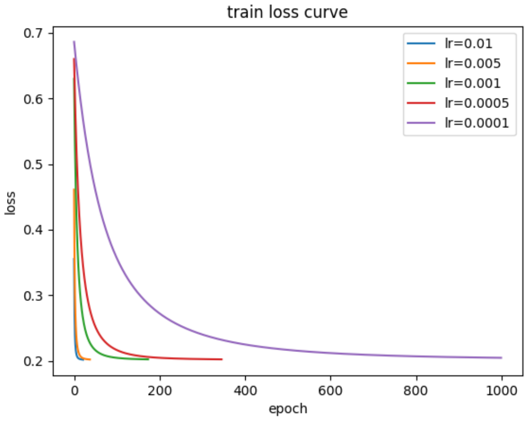

Figure 2: Training loss curve

We decide to use a learning rate of 0.005, batch size of 5000, and epoch number of 100 for the remaining training. Table 2 displays the metrics of the model on the validation set.

We set the cutoff to 100 and apply the model to predict the relevance scores of passages for each query in the file *candidate_passages_top100.tsv*. We save the top 100 highest-scoring passages for each query in a file called *LR.csv*.

Table 2: Metrics of LR model

LR	Cutoff	mAP	mNDCG
Top 100	100	0090	0.1254

3 Task 3: LambdaMART Model

3.1 Model

3.2 Training

3.3 Hyper-parameter Tuning