E-commerce Sites Search Results Relevance Prediction Based on Ensemble Approach

Qiqi Wang
School of Economics and Management Department, Beihang University, Beijing 100083, China;
wangqiqi@buaa.edu.cn

Keywords: Search Relevance, E-commerce site.

Abstract. Though there are numerous traditional models to measure the search relevance of search engine, the evaluation results of existing models are not precise enough and difficult in operation in most of the cases. This article proposed a forecasting approach based on ensemble method to improve the precision of search relevance prediction. The critical process of the approach requires features extraction and parameters selection. The experimental results show that it works well in given data set. Furthermore, in order to select the optimal weights set, we also give a model and algorithm for the problem in the end of the article.

1. Introduction

Search relevance evaluation is a challenge for both Industrial and academic researchers. The problem of predicting the relevance of search results is a hot topic in information retrieval (IR) field. In IR, given a query, there are some terms ordering in rank given by the algorithm in search engine. However, this work does not consider the position of search results. Given a query corresponding to a relevance score (1 to 4) rated by human beings, as well as some features extracted from the queries, titles, descriptions and their similarities, we try to use supervised method to ensemble several classifiers into a model for search results prediction. In addition, we conduct an extensive experimental evaluation experiment using neural networks for further exploration. In the end, the article also proposed an algorithm for the weights selection in ensemble model.

2. Related Works

2.1 Ranking Relevance.

As for search relevance ranking, most of the previous related works use text matching and click modeling [1]. For instance, the user behavior model has been explored by many researches [2]. Dawei Yin et al. [1] list the up-to-date methods and what Yahoo had adopted in ranking relevance. The work took refreshes, query features, topical matching, text matching, document statistic, time and location as well as other variables into consideration. In this work, we try to explore features of queries, titles, descriptions and their similarity features as much as possible to achieve a better prediction result. The method in this work can also be used in other fields such as NDCG.

2.2 Evaluation Methods.

There are many indexes for evaluating search results based on queries, such as precision, recall, BM25 and NDCG. Precision and Recall are two traditional evaluation methods in information retrieval. Precision is the fraction of the retrieved documents that are relevant presented as $P = \frac{R_a}{R}$.Recall is the fraction of the relevant documents that are retrieved presented as $P = \frac{R_a}{R}$. F1 measure is a harmonic mean of Precision and Recall presented as $P = \frac{2PR}{P+R}$. However, it is difficult to estimate the actual precision and recall in a system, since it requires all the relevant documents known. Obviously, it is unnecessary to acquire all the relevant documents in the search results. BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in ach document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity)[3]. What should be noted is that BM25 requires the total number of

documents in the collection to be known. NDCG refers to normalized DCG. It evaluates the results by the ranks scored by raters and the position in the result [4].

In this work, we only take the rankings scored by raters into consideration. Therefore, we compare the results the model output and the real results given by human beings. The evaluation index used in the experiment is quadratic weighted kappa, which is a measure of inter-rater agreement between two raters that provides discrete numeric ratings. In machine learning it can be used to quantify the amount of agreement between an algorithm's predictions and some trusted labels of the same objects [5]. Potential values range from -1 (representing complete disagreement) to 1 (representing complete agreement).

3. Problem Definition

Currently, small online businesses seldom have good way of evaluating the performance of their search algorithms, which is difficult for them to provide an exceptional customer experience.

The goal of this paper is to create a model that can be used to measure the relevance of search results, which means that the model should get results as similar to the scores given by human beings as possible, so as to predict the ranks as precisely as possible. To be brief, the challenge in this paper is to predict the relevant score given the queries, product descriptions, product titles, median relevance, relevance variance and rank labels given by human raters.

4. Model

The problem of predicting the search relevance can be translated into the problem of classification of ranks. The four ranks can be viewed as four class labels. Based on these assumptions, we build an ensemble model by combining separate classifiers with different weights.

4.1 Ensemble Models.

This part introduces an ensemble of six models, which are SVM, Adaboost, GBDT, random forest and two other models based on extracted TFIDF features respectively. By calculating the kappa loss function on the weighted predictions in 5-fold cross validation, we got benchmark in each separate model. The ensemble uses a simple linear weighted combination of the model predictions, and it is weighted by the relatively optimal weights, and then using simple rounding to the resulting median relevance so it always takes the values of 1, 2, 3, or 4. The ensemble process acquires the optimal weights to use, which will be discussed in the end of the article.

The framework of the work can be illustrated in the flow chart below.

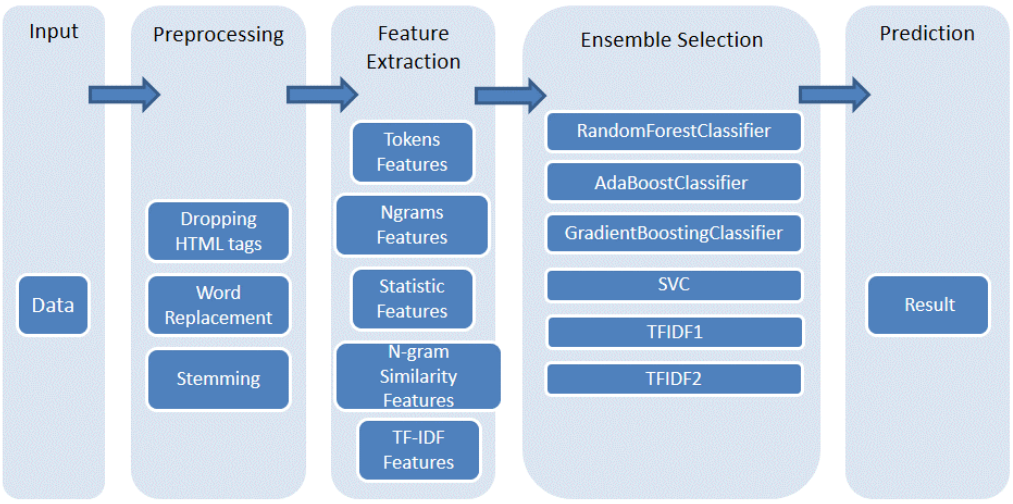

Fig. 1 Flow Chart of the Work

The measurement of the separate model and the ensemble model use the quadratic weighted kappa

$$K=1-\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}O_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}E_{ij}}$$

where

$$w_{ij}=\frac{(i-j)^2}{(N-1)^2}$$

O_{ij} Corresponds to the number of search records that received a rating of i by A and a rating of j by B, and E_{ij} corresponds to the matrix of expected ratings.

4.2 Neural Networks.

In this section, we performs an experiment on the training set with a three layer Feed-Forward Neural Networks. There are twenty eight nodes in input layer, seventy nodes in hidden layer, and a node in output layer. The calculation is conducted with the aid of the Pybrain module of python.

Figure 2 shows the network structure we designed.

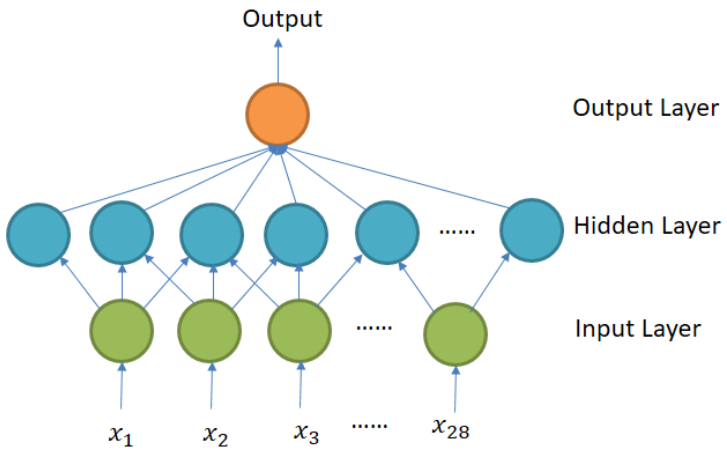

Fig. 2 A Three-Layer Feed-Forward Neural Networks

The input layer and the output layer uses the linear function, and the hidden layer uses the sigmoid function, which proved to be of the optimal matching by experiments in section 5.

5. Experimental Study

5.1 Data Description

The data we use include the training set and testing set. The fields in the training set includes product id, search terms used, the full product descriptions along with HTML formatting tags, median relevance score by 3 raters and variance of the relevance scores given by raters, which are titled id, query, product description, median relevance and relevance variance separately.

The testing set only includes id, query and product description. The data provided in the product description field is raw and contains information that is irrelevant to the product. To ensure that the algorithm is robust enough to handle any noisy HTML snippets in the wild real world, the preprocessing is needed to clean up the data.

5.2 Preprocessing

The preprocessing process includes HTML tags dropping, word replacement including spelling correction, synonym replacement, stop-words removing and raw dataset stemming. Besides, dimension reduction and standardization also performed before modeling.

5.3 Feature Extraction/Selection

The features extracted in the experiment can be classified into several categories as illustrated below.

5.4 Tokens Features

The features related to tokens include intersection rate between query and title, query and description, as well as the length of query, description and title.

5.5 N-grams Features

The features related to n-grams features include two grams in query, title and two grams in query and description.

5.6 Statistic Features

Since there are lots of identical queries in training set, the queries are grouped for facilitate of statistic. The statistical features include mean training relevance, median of mean relevance in training set and average relevance variance.

5.7 N-gram Similarity Features

The n-gram similarity features include average one gram similarity between titles and grouped ratings, average two grams similarity between titles and grouped ratings, as well as average one-gram similarity between descriptions and grouped ratings, average two-grams similarity between descriptions and grouped ratings.

5.8 TFIDF Features

The TFIDF features include vectorized TFIDF values for concatenated query, title and description in each record.

5.9 Dimensionality Reduction

The bag of words TFIDF model build a matrix with high dimension. To improve training efficiency, dimension reduction should be conducted before fitting into a model. PCA (Principal Component Analysis) and SVD (singular value decomposition) are two common methods for dimensionality reduction. Contrary to PCA, SVD does not center the data before computing the singular value decomposition [6]. This means that it can work with scipy.sparse matrices efficiently. Consequently, the experiment adopts the method of SVD to perform dimension reduction.

5.10 Cross Validation

The number of records in training set is 32136. To facilitate the testing of model, 1563 records are selected as initial data set. These data are split into training set and testing set with the proportion of 8:2. Then the training set is split into two parts for training and validation respectively. The experiment performs 5-fold cross validation. In the following experiment, the training set is expanded gradually. It turned out that when the size of training set increased to 6020, the score improved 10%.

5.11 Results and Measure

This section describes the results of different separate models, and presents the measure method of the result. It is well known that the mean squared error (MSE) is a common index for measuring the difference between the estimator and what is estimated. In the continuous output, the index can work well. However, the result we get is discrete, in which condition, the kappa based index is superior to the MSE index. Therefore, we use kappa based method to measure the agreement between the score from human raters and the predicted result.

5.12 Ensemble models

Given initial weights set and classifiers set, we got an agreement score of 63%. The corresponding weights are list in Table 1.

Table 1. Optimal Weights of Classifiers

classifier	weight
Random forest classifier	0.5
SVC classifier	0.1
Adaboost classifier	0
GBDT	0.1
TFIDF1	0.1
TFIDF2	0.2

With the ensemble method, we got the score of 5-fold in each model which can be seen in Table 2.

Table 2. Scores of Classifiers in 5-Folds Cross-Validation

	Random forest classifier	SVC classifier	Adaboost classifier	GBDT	TFIDF1	TFIDF2
1	0.604	0.540	0.560	0.564	0.526	0.504
2	0.596	0.493	0.546	0.548	0.513	0.535
3	0.567	0.474	0.498	0.519	0.541	0.523
4	0.561	0.513	0.516	0.544	0.520	0.569
5	0.590	0.527	0.581	0.602	0.533	0.485
Average	0.584	0.509	0.540	0.555	0.527	0.523

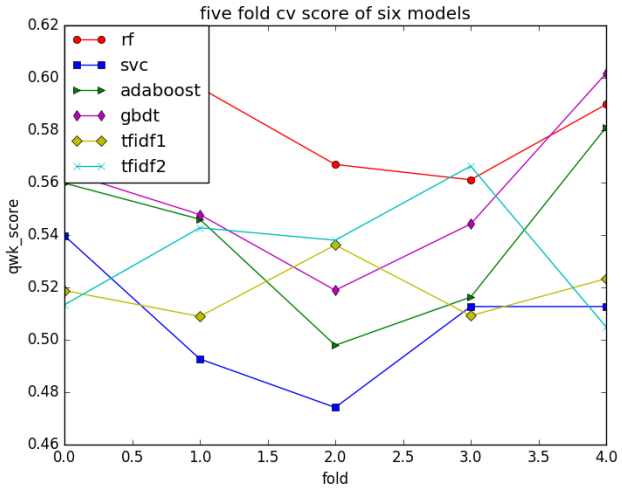

Fig. 3 Five-Fold Cross Validation of Six Models

As can be seen in Figure 3, the random forest achieved optimal score in separate models. In addition, the SVC classifier performs worse than other classifiers. According to the performance in single model, we can adjust the weights of each model in the ensemble process.

In the ensemble models, the score achieved 63%, which exceeded performance of each separate model, and thus the result show the significance of ensemble.

5.13 Neural Networks

In this section, we explore the data with a three-layer neural network. By performering experiment, we find the best matching among different types of layer. Finally, we set both of the input layer and output layer as linear function, and the hidden layer is sigmoid function.

By performing experiment, we found that it achieved about 60% of the agreement between the target and prediction result. In the following experiment, we also use the RNN, however, it did not perform as well as the Feed-Forward Neural Network.

5.14 Further Discussion

It should be noted that in the ensemble model, we give initial weights set as a constant. In the experiment, we find that different weights set achieves different scores. The influence can be varied from 1% to 5%. Therefore, in the section, we want to find a method for selecting the optimal initial weights set.

Consider the formula of quadratic weighted kappa, we define f(w) as the function of w. Our goal is to make the score maximum under the constraint of w. The model can be represented as follows:

$$\text{Min } \sum_{j=1}^{n}(|y - z|) \quad (1)$$

$$\text{s.t. } 0 \le w_i \le 1 \quad (2)$$

$$\sum_{i=0}^{n} w_i = 1, i = 1,2,3 \dots \dots n \quad (3)$$

where n refers to the number of models in the experiment.

The objective function is nonlinear, and the constrains have both equation and inequation. To solve the model, we can adopt the iteration algorithm as follows

Table 3. iterated algorithm

Algorithm
1. Assign values to w within [0,1] based on the performance in the separate model
2. Repeat
for $w_i \in$ w do
count $\leftarrow$ 0
$g_{i,t} \leftarrow$ solution of (1) to (3)
if $g_{i,t} - g_{i,t-1} <$ θthen count++
$g_{i,t-1} \leftarrow g_{i,t}$
until count=n
return g_{t-1}

The validity of the algorithm remains to be test. In the next stage, we will find the best weights by the algorithm and check the validity of the algorithm.

6. Conclusions

The article explored the raw data set with plain texts and labeled ranks by human raters. Based on the data, we performed preprocess, feature extraction and ensemble modeling successively. The results indicated that the ensemble method achieves better scores than separate models. Then we adopted a three layer neural network and adjusted the parameters to observe the agreement results. The difficulties in the experiment are feature extraction and how to convert to the features vector to fit the training models, as well as the determination of weights set, which will be explored in the future work. The feature based method is also of significance, and it can provide a basis for the calculation of NDCG, which is widely used for evaluating ranking relevance in information retrieval.

Acknowledgements

The work is supported by national natural science foundation of China (NSFC) under 71571006.

References

[1]. Yin, D., Hu, Y., Tang, J., Daly Jr, T., Zhou, M., Ouyang, H. & Langlois, J. M. Ranking Relevance in Yahoo Search.

[2]. E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In SIGIR ’06

[3]. Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM 25 and beyond. Foundations and Trends® in Information Retrieval, 3 (4), 333-389.

[4]. Al-Maskari, A., Sanderson, M., & Clough, P. (2007, July). The relationship between IR effectiveness measures and user satisfaction. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 773-774). ACM.

[5]. Viera, A. J., & Garrett, J. M. (2005). Understanding inter observer agreement: the kappa statistic. FAM Med, 37 (5), 360-363.

[6]. Xu, P. (1998). Truncated SVD methods for discrete linear ill-posed problems. Geophysical Journal International, 135 (2), 505-514.