Web Search Relevance Improvement
Shijia Bian
STA 571 Final Project

Introduction

The searching engine is widely implemented on various types of websites. The customers expect the searching engine to give them the most relevant result matching their inputs. The business also keeps improving the accuracy of the searching result in order to provide the customers with good user experiences, and also compete with the their competitors. However, it is difficult to have the searching engine to give the accurate searching results all the time, especially when the inputs are not informative. The traditional way of improving the searching experiences is to manually label those inputs that are not able to result the accurate results. This is very expensive and time-consuming. In this project, I am going to summarize the topic modeling and ranking algorithm. Then, I am going to create a new algorithm by combining these two techniques. The topic model is applied to be an initial filter (feature generation) and return the specific topic the inputs might belong to, and then the ranking algorithm can give the relative relevance for each results under this topic. Other possible ML algorithm will be possibly tried in the final paper, but these two algorithms are the main algorithms used to develop the learning method in this paper. The experiment will be carried on the data provided by the "Home Depot Product Search Relevance" on Kaggle.

Related Work

The web searching relevance ranking has been studies by many academic institutes and companies over the past decades. The website has a list of the results that might be relevant to the users' input queries[1]. Then the website will return this list of result ordered by the relevance score: the most relevant result comes first. The main internal ranking algorithm is open to the public, but the more detailed technique remains hidden, so the company can compete with the other competitors[1]. The good search ranking algorithm can not only return the accurate results in terms of the customers' needs, but also can provide the relevant results that can be novel and surprising for the customers. These algorithms includes the traditional ML algorithms, deep learning techniques, collaborative filtering and so on [2]. The Netflix competition is a well-known example. In addition, Google, Amazon, Facebook and other companies have spent years in improving the web searching relavance performance [3].

We are assuming that the training data contains the customer inputs, the returned result, the manually defined relevance score and the detailed description for the returned result. The customer inputs are strings of words. The returned results are item names that are also strings of words. The relevance score is a numeric score, but it only has 13 options. It ranges from 1 to 3 with 0.25 differences. The descriptions are a string of texts describing the returned items. In this supervised learning, we are going to build up a model that can classify these data set in terms of the relevance scores that are like labels. Thus, the input variables are the customers' inputs and the extracted features from the item descriptions. According to the exploratory analysis, the most difficult part is for all the the searching inputs can be found in the item name or item description, but their relevance scores are still not full score and have difference. It is very difficult to build up a sensitive model to detect this difference. Therefore, the combination of the topic model and ranking algorithm can help. The returned results are considered to be topics. The features motivated by topic modeling idead can preliminarily define if the returned results are topics for the returned results. Then the ranking algorithm can give a relevance score. We are expecting that this model can evaluate the the test set and give the accurate relevance score for the customer inputs.

The two main algorithms we are going to adopt are the topic modeling and ranking algorithm.

The topic model serves as a preliminary filter before running the ranking algorithm. The way the filter works is providing a score that can be used for the next step of learning. So we want first to conduct the feature extraction. The feature extraction needs to be conducted for three parts: the item description, the input query and the item name. This will first eliminate the stopping words in these three parts. The key words can indeed be extracted, such as the product name, usage, size, special instructions and so on. Then we want to calculate the frequency of the key words appearing in the description and item name. Therefore, we have these situations below: 1). the searching key words do not appear in the keywords; 2) the searching key words partially appear in the keywords, such as $1/2, 1/3, 1/4, \ldots$ of the searching keywords appear in the keywords; 3) all of the searching key words appear in the keywords; 4). The frequency of the corresponding keywords appearing in the description and the product name. Therefore, we can build a model to calculate the probability that the returned result falls into the same category in terms of item name and the description. This probability will be considered as a score for the next step of ranking modeling.

The ranking algorithm will take the relevance score as a response variable. We will regress this response variable on the extracted features from the last step. So we can see that the probability of getting a specific score in terms of the available features. To simplify, we can use the algorithm, random forest, to automate the process.

After training this model, we can test the model in the test data set. The model will output the probability of getting each score. So we can choose the relevance score with the highest probability.

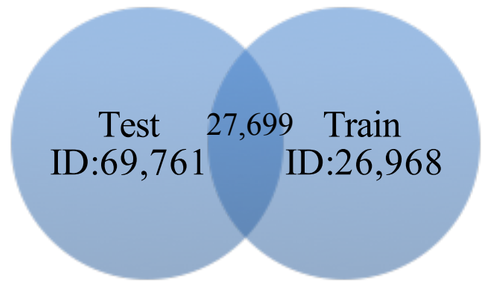
The Case Study on Home Depot

- **Data**
  - Train Data: 74,067
  - Test Data: 166,693
  - Given Information:

	id	product_uid	product_title	search_term	relevance	product_description
0	2	100001	Simpson Strong-Tie 12-Gauge Angle	angle bracket	3.00	Not only do angles make joints stronger, they ...
1	3	100001	Simpson Strong-Tie 12-Gauge Angle	l bracket	2.50	Not only do angles make joints stronger, they ...
2	9	100002	BEHR Premium Textured DeckOver 1-gal. #SC-141 ...	deck over	3.00	BEHR Premium Textured DECKOVER is an innovativ...
3	16	100005	Delta Vero 1-Handle Shower Only Faucet Trim Ki...	rain shower head	2.33	Update your bathroom with the Delta Vero Singl...
4	17	100005	Delta Vero 1-Handle Shower Only Faucet Trim Ki...	shower only faucet	2.67	Update your bathroom with the Delta Vero Singl...
5	18	100006	Whirlpool 1.9 cu. ft. Over the Range Convectio...	convection otr	3.00	Achieving delicious results is almost effortle...

- Goal: predict the relevance score for each search_term in the test data set.

- **Data Visualization [4]**



- **Challenges:**
  1. overlapping of the products ID from the test data set and the training data set is small;
  2. the slight different searching query can result in different scores under the same product;
  3. the searching queries that containing the same key words can result in different/same scores for different products, the extraction of the description word is important;
  4. The exact matching of the word might lead to overfitting.

- **Adding New Features to Replect the Topic Model**
  1. the number of word match from the searching query to the the product name;
  2. the number of word match from the searching query to the the product description;
  3. the proportionof word match from the searching query to the the product name;
  4. the proportion of word match from the searching query to the the product description;

  .....

- **Relavance Ranking Algorithm**
  1. Rnadom Forest
  2. XGBoost

- **Criteria: MSE**

The performance has been improved from 74% to 70%. More sophisticated features are expected to be extracted and using Xgboost for the relevance score predicting, hit 51.050% benchmark for this Project.

[1] H. Zaragoza and M. Najork. *Web search relevance ranking*. In L. Liu and M. T. Ozsu ¨ (editors), Encyclopedia of Database Systems, pages 3497–3501. Springer US, 2009.
[2] Berberich, Klaus, and Dhruv Gupta. *Recommender Systems*. Max Planck Institut Informatik. Saarland University, 2014. Web.
[3] Greg Linden , Brent Smith , Jeremy York, *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*, IEEE Internet Computing, v.7 n.1, p.76-80, January 2003 [doi>10.1109/MIC.2003.1167344]
[4] Brian Carter. *HomeDepot First Data Exploration*. Home Depot Product Search Relevance. Kaggle.com, 19 February 2016. Web.