# Midterm Report

Ehsan, Shifu Xu, Yu Wen

# Changes

We didn’t make any changes in our objective, or data;

Instead of using the methods in the proposal, we use the random forest to train our model, details please refer to section 3.

2. Data preprocessing

(Describe data preprocessing and why. This includes data cleaning, selection, feature generation and representation, etc.)  
 Since “test.csv” is only used for prediction, and contains no true labels(in our case, the “relevance score”) , we firstly create a development set “developmentTestSet.csv” by randomly select 20% data from “train.csv”; As for the remaining 80% data, “subTrainSet.csv”, we use it as our actual training data , this random sample selection was implemented in DevelopmentSet.java.

Then we compute the tf-idf with regards to product title and product description as our features.

As we can see, the most intuitive way to regards a “query” as more “relevant” to one product is that the query words have much “similarity” in the product title, which can be a good represent of one specific product. Intuitively, we can count the number of words of one query which appear in one specific product title (suppose we are interested in calculating the relevance score between this query and this product with this product title), but this have disadvantage: some words in the product title appears more than one time, indicating that if in one specific query, this word appears, we should give this query more “attention” since it has some “more important” word, however, only counting the number of words of one query which appear in one specific product title can’t achieve this.

As a result, we can use the term-frequency to indicate this relationship. Firstly, we have to extract the “interested item” of the product tittle such that we should remove the unrelated words appear in the product tittle such as “a”, “the”, ”for” and etc. which didn’t have any realistic meaning, which are also defined as stopping words. We achieved this preprocess by using the class “CountVectorizer” in “sklearn” package.

Since when we considering using product tittles to represent one product, we only have to consider different “interested items” appear in all the different product tittles. So we can treat all the product tittles as the corpus, which can be used to extract the vocabulary (the “interested item” as mentioned before). As long as we have this vocabulary of product tittle corpus, we can represent every product tittle as an vector : v=(v1, v2,..vn), n=the number of vocabularies, the component vi stands for the number of times vocabulary i appears in v; Also, we can count the

3. Methods tried  
-- what methods you have tried towards the project goal? and why do you choose the methods?  
 we use the random forest for

4. Evaluation

for now

-- evaluation: what results you have achieved up to now? what is working or what is wrong with the model?

5. next step  
-- next step