Today we discuss: can a digital rater works better than a human? It is a Machine learning problem in natural language processing field, with home depot input data.

There are 6 parts we goanna go through today one by one, so 1st project purpose

The algorithm focus on the prediction of relevance of search results to fast pass the correct item to their clients. So the speed and accuracy is two major factors to evaluate different smooth model and optimization. The most optimal although can achieve 0.474 RMSE for potential errors with random forest and word2vect algorithm, which I goanna detail introduce later.

This search algorithm will make an impact on both of the monetary side for the whole company, as well as the project management side.

**Search algorithm as project manager**

Home Depot works on multiple software projects. A common challenge is that problematic issues are always discovered after providing the item to the clients, which then call for post-mortem investigations to determine the root cause. The search algorithm, a pro-active solution would save time and reduce the risk in the first place.

Also, the algorithm can **save employee cost** and increase the revenue by reducing the number of human raters.

Our team decided to address this problem by finding most accurate analytical approach.

linear regression and random forest are two ensemble learning method for regression problem, to measure the relation among multiple variables.

Linear regression works well when all the features in the input corpus are truly linear and independent variables. Otherwise, non-linearity transformation will be required to transfer to a strictly linear data, or strong linear component. Also, it is a good supervised and simple learning algorithm when a great number of the features are given. A clear disadvantages of linear regression is the high error, so we just get 0.478 here, compared to random forest. And, of course, when the corpus feature is not linear but dependent, LR will not be the best choice.

RF is good on **reduction in overfitting**: by averaging multiple trees with low bias, there is a significantly lower risk of overfitting.

It is great for **less variance**: By using multiple trees, the algorithm reduces the chance of stumbling across a classifier that doesn’t perform well because of the relationship between the train data and test data. And the algorithm selects a random subset of features when splitting to reduce the correlation between the trees to get extra lower variance. So it is more accurate and flexible and Input friendly to handle large proportion of missing data in the input corpus.

As the error decrease to 0.474, we select random forest as our final strategy. But this model is a little slow and complex compared to linear regression.

For word embedding algorithm, we pick Word2vect which decrease the error from 0.49 to 0.48.

match degree is based on semantic context, to guarantee most of the classification is logically make sense, so the synonyms can be grouped in similar feature and classification. This algorithm is widely-used in machine learning and deep learning model with high compatibility in supervised learning method. This means, human interaction and label is required. If too many unknown vocab, or out-of-vocabulary words given in the input data, and the model hasn’t encountered the word before, it will have no idea how to interpret it or how to build a vector for it. Then the algorithm forces a random vector which is not ideal case we want to. It is a little bit slow and not work for polysynthetic language such as Arabic, German or Turkish in unsupervised learning without label provided.

The base pipeline of the model is only 0.494 computed on common count, fuzzy score, and the length of search keyword.

It is the pipeline result with and without word2vec, linear regression and random forest. The linear regression always get higher error than random forest in the same situation. So the error in The final strategy is 0.474 performed by random forest and word2vec with base line algorithm.

to explore the error above, we take two major cases into consideration.

Phrase can be tricky to count, for instance, while searching “fuel oil”, the relevance has been over count by word “oil”, which is not exactly the same as search keyword but still count into the model. This lead a difference between the real relevance,1, and the predicted relevance increase to 1.404.

Besides phrase, semantic error also draw us attention. When searching “plastic case”, we got both of “plastic” and “case”, which is difference than the first situation, but from two different source instead of one source contain “plastic case” together, where “plastic” refers of “plastic” bag and ”case” as a unit. The real relevance stay at 1, but the difference between the real and predicted as 1.402.

Overall, this strategy is constructed by random forest and word2vec.

We also recommend Home Depot pay more attention on the length of search keyword, and context of product title and description over the corpus. These are the most important text context will influence the relevance and regression result. Since the match degree is highly based on the similarity between search keyword and product title, as well as search keyword and product description.

In the future, our team will focus on spelling check, which may reduce some potential error and increase the accuracy.

Grid Research is another potentially functional algorithm when adjusting the parameters. Since we are doing it right now manually, a systemic grid search will be helpful to find the optimal parameters and reduce the error. Also Deep learning RNN which is known for classical text learning method with snorkel to de-noise the data.

Thanks you!