Machine Learning Engineer Nanodegree
Capstone Proposal

Shun-Wen Chang
April 7th, 2017

Quora Question Pairs (from Kaggle[1] Competition)

Domain Background

Quora is a question-and-answer website where people ask all kinds of questions and they are answered or edited by its community of users who happen to have the knowledge in the areas. It’s a great way to share knowledge, experience, and opinions across the vast group of people around the globe. Its publisher, Quora Inc.[2], is founded in 2009 and based in Mountain View, California.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly questions. Multiple questions with the same intent can cause seekers to spend a lot of time finding the best answer to their question. However, we don’t want to eliminate all similar questions because this is a good way for users to know if their questions are common or not. What we need is an algorithm to group all duplicate questions together and list them out when a user is seeking the answer to that question. Those who want to answer the questions can do it all at once, too. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

Problem Statement

This is a binary classification problem. Inputs are two sentences of texts and the goal is to predict if these two sentences are of the same meaning.

I will be tackling this as a natural language processing problem and use TF-IDF vectorization[3] to process input texts. Then I plan to use techniques such as regression and decision trees to train the dataset. The features will be extracted from sentences such as word count, character count, and word distribution. The target here is either “is duplicate” or “not duplicate” between pairs of questions.

Datasets and Inputs

The datasets are provided by Quora on Kaggle competition website. They are free to download.

Input Data fields

- id - the id of a training set question pair
- qid1, qid2 - unique ids of each question (only available in train.csv)
- question1, question2 - the full text of each question
- is_duplicate - the target variable, set to 1 if question1 and question2 have essentially the same meaning, and 0 otherwise.

Solution Statement

The solution will be predictions of either duplicate or not in the test dataset. First I will use TF-IDF to process all the texts and do some visualization of the data to get some understanding. Then I will perform feature extraction and select features such as word length, word count distribution, character count.

For training models I will compare logistic regression, decision trees, nearest-neighbors, and SVM since this is a classification problem. Finally I will select the best model for this problem and fine tune parameters to get best accuracy.

Benchmark Model

For this problem, the benchmark model will be random forest model. I will try to beat its performance with other algorithms.

Evaluation Metrics

Prediction results are evaluated on the log loss between the predicted values and the ground truth. According to Kaggle competition webpage, the ground truth is the set of labels that have been supplied by human experts. The ground truth labels are inherently subjective, as the true meaning of sentences can never be known with certainty. Human labeling is also a 'noisy' process, and reasonable people will disagree. As a result, the ground truth labels on this dataset should be taken to be 'informed' but not 100% accurate, and may include incorrect labeling[1].

Since this is a Kaggle competition project, I will take the leaderboard score as my evaluation.

Project Design

Before even start training models, I will first take glimpse of the data see what the shape and is and how they are formatted. Then I will start doing my natural language processing and extract information such as character counts, sentence length, TF-IDF vector...etc. Since in this case there are not too many features, I don't think PCA[4] feature selection is required. I may perform some graph visualization for better understanding of the data distribution. This depends on whether I can find such existing implementation/library or whether I have enough time to do it from scratch.

To train models, I plan to choose 3-4 different models to compare. Because this is a classification problem, a few approaches in my head would be regression, decision trees, SVM, KNN, and random forest. Using cross-validation I can find which model performs best, and then use that one to tweak relative parameters.

I expect to spend 60% of the time on data cleaning and natural language processing part and 40% of the time on training models and tweaking parameters. The final accuracy will be calculated against the test data set provided by Kaggle.

Reference

1. https://www.kaggle.com/c/quora-question-pairs
2. https://www.quora.com
3. http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
4. https://en.wikipedia.org/wiki/Principal_component_analysis