Machine Learning Engineer Nanodegree
Capstone Proposal

Shun-Wen Chang
April 6th, 2017

Quora Question Pairs (from Kaggle Competition)

Domain Background

Quora is a place to gain and share knowledge about anything. It’s a platform to ask questions and connect with people who contribute unique insights and quality answers. This empowers people to learn from each other and to better understand the world.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers, and offer more value to both of these groups in the long term.

Problem Statement

The goal of this project is to predict which of the provided pairs of questions contain two questions with the same meaning. I will be tackling this as a natural language processing problem and apply advanced techniques to classify whether question pairs are duplicates or not. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

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 and SVM since this is a binary classification problem.

Benchmark Model

For this problem, the benchmark model will be a *x* percent chance prediction, where *x* is the proportion of "is duplicate" among all question pairs in the training set. This is a very naive way and therefore serves as the benchmark model. Whatever I'm trying to achieve should be higher than this value, otherwise it means my model is not good enough.

Evaluation Metrics

Prediction results are evaluated on the log loss between the predicted values and the ground truth. 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. We believe the labels, on the whole, to represent a reasonable consensus, but this may often not be true on a case by case basis for individual items in the dataset.

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 PDA 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 2-3 different models to compare. Because this is a classification problem, a few approaches in my head would be regression, decision trees, SVM, 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

- Kaggle - https://www.kaggle.com/c/quora-question-pairs
- Quora - https://www.quora.com
- sklearn TFIDF - http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html