Machine Learning Nanodegree

Capstone Proposal

Peter Herr
May 15 2017

Domain Background

The project domain I am pursuing is in the area of data deduplication & matching, and Natural Language Processing (NLP).

- In the domain of master data management, one of the key components is identifying unique or master records so that a common reference point is available for sharing and use by other applications or end users. This allows the organization and it's users to minimize confusion that duplicate records cause, and to focus on data quality for use downstream.
- In the domain of Natural Language Processing, the goal is identify patterns, insights, and/or meaning through the processing of natural language (speech, signing, writing). This is a vast domain, and it serves quite a few purposes. Example tasks for solving NLP problems include sentiment analysis, topic segmentation, part-of-speech tagging, and even speech recognition (among many others).

I am personally interested in these two domains above, as it relates to a problem my current clients face in the music royalties/copyright industry, where it's critical to identify master records and relationships between entities in order to efficiently attribute the proper works to their owners. I'm also interested in how these topics can be used to increase the efficiency and automation of many enterprise systems, such as CRMs.

References:

- http://bit.ly/2qm2Y6c
- http://bit.ly/22KVy5T

Problem Statement

The project I am pursuing is a Kaggle Competition titled “Quora Question Pairs”. The question being posed is: “Can you identify question pairs that have the same intent?”.

Per the introduction, the problem statement is as follows: *The goal of this competition is to predict which of the provided pairs of questions contain two questions with the same meaning. 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.*

One solution, as mentioned, is the use of manual human tagging. A second solution, as shared by Kaggle, is that Quora currently uses a Random Forest model to identify duplicate questions.

Datasets and Inputs

Quora has provided a training and test datasets containing the full text of question pairs, and whether or not these two questions were considered as duplicate. The train.csv.zip dataset contains following 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.

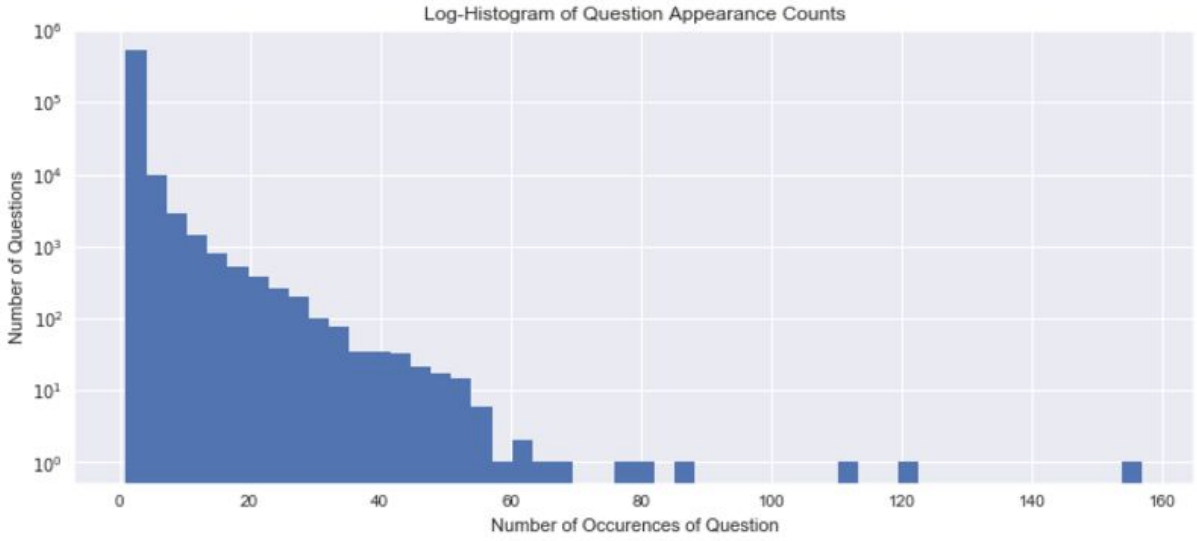
Here’s the first 5 records from the training dataset:

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

The test.csv.zip dataset contains three fields: **test_id**, **question1**, and **question2**.

The training dataset contains 404,288 rows and 6 columns, and the test dataset contains 2,345,796 rows, and 3 columns. Within our training dataset, we have 255,027 (63.08%) non duplicates, and 149,263 (36.92%) duplicates.

In reviewing individual questions that appear multiple times, we can see that across both our testing and training datasets, there is a total of 537,933 unique questions, and 111,780 of these questions appear more than once.



As visible in the histogarm above, the vast majority of questions either appear once or a few times. Additionally, there is a very small number of questions that appear up to around ~50 times, and three questions that appear over 100 times.

The submission that Kaggle scores should be provided in two columns: test_id, and ‘is_duplicate’. Additional details can be found here: http://bit.ly/2qLenhu

Solution Statement

This is essentially a classification problem (0 = not duplicate, 1 = duplicate). A solution to this problem is to find patterns of information present between the two questions being compared, and to use this information to determine whether or not these are duplicate or not.

After reviewing the data, I will explore approaches to clean the data (remove punctuation and stop words) and to generate new features. I believe that generating new and meaningful features from the question pairs is likely one of the most important

approaches to solving this problem. I will explore the use of NLP techniques to generate new features such as:

- **Shared Words:** Review the percentage of words shared by the two questions
- **Term Frequency - Inverse Document Frequency (TF-IDF):** a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus (wikipedia). Term frequency is essentially quantifying the frequency of a given word or term within a corpus. Inverse Document Frequency is a measure of how much information the word provides (whether common or rare).
- **Levenshtein Distance:** a string metric for measuring the difference between two sequences (wikipedia)
- **Feature Extraction using CountVectorizer and TfidfVectorizer:** CountVectorizer (sklearn) converts a collection of text documents to a matrix of token counts. TfidfVectorizer (sklearn) converts a collection of raw documents to a matrix of TF-IDF features

Next, I will use classification algorithms such as Logistic Regression (sklearn) and XGBoost (github) to generate predictions leveraging my training data and newly generated features. In terms of measuring and quantifying the success of my approach, I will use the log loss function to quantify the accuracy of my predictions on the training and test sets.

Benchmark Model

Currently, Quora uses a random forest model to identify duplicate questions. For my benchmark model, I plan to use a random forest model as well. I plan to measure the random forest model I create using the log loss metric. I'll be measuring using this metric, since that is also what I plan to evaluate my results on (and it's also what Kaggle is scoring submissions on).

Evaluation Metrics

In this particular competition, submissions are evaluated on the log loss between the predicted values and the ground truth.

This is a good evaluation metric as it quantifies the accuracy of a classifier by penalizing false classifications. When we minimize log loss we are essentially maximizing the accuracy of the classifier.

According to fast.ai (http://bit.ly/2qpAuqB), the definition of Log Loss is as follows:

Logarithmic loss (related to cross-entropy) measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The goal of our machine learning models is to minimize this value. A perfect model would have a log loss of 0. Log loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high log loss.

Here's an example of the mathematical representation of the Logistic Loss function:

$$V(f(\vec{x}), y) = \frac{1}{\ln 2} \ln(1 + e^{-yf(\vec{x})})$$

Project Design

My approach to this project includes a number of steps below which I will work through, and iteratively return back to as-needed in order to make adjustments to my approach based on findings:

Exploratory Data Analysis - My first step after loading the data will be to explore the dataset and try to understand the nature of the data. This will involve activities like understanding how many question pairs are duplicate/not duplicate, how many individual questions are repeated across the dataset, and how many words and characters are in common between those questions that are duplicates vs those questions which are unique.

Text Cleaning - In addition to the above, I am going to research methods for cleaning questions to hopefully be able to make better comparisons across question sets. This involves considering the following: removing stop words, correcting any spelling mistakes, and considering the removal of punctuation.

Generating New Features / Feature Engineering - I believe one of the key approaches to solving this problem will be by generating new features based off of the unstructured text that makes up each of the questions.

Classification - Since the goal of this project is to determine whether or not a pair of questions is duplicate or not, we are looking to solve for binary classification. Thus, I

can solve this using a classifier. I plan to attempt using XGBoost and other classifier algorithms to solve this problem.

Natural Language Processing - Natural Language Processing is an area of interest for me, and so depending on how much effort it takes me to build out the model and approach using the above steps, I plan to explore opportunities to improve my model using NLP techniques.

To support my learning as I progress through this project, I plan to leverage the resources available through the Kaggle Competition (http://bit.ly/2mGjD01) as well as elsewhere on the web.