Machine Learning Engineering Nanodegree

Capstone Proposal

M Rao
08/14/2018

Quora Questions – identifying question pairs that have the same intent.

Domain Background

Since 1995 the Internet has tremendously impacted culture and commerce, including the rise of near instant communication by email, instant messaging, telephony (Voice over Internet Protocol or VoIP), two-way interactive video calls, and the World Wide Web with its discussion forums, blogs, social networking, and online shopping sites.

The Internet continues to grow, driven by ever greater amounts of online information and knowledge, commerce, entertainment and social networking. This has introduced many websites to be built that mostly cater as a resource to a user.

Quora is a question-and-answer site where questions are asked, answered, edited, and organized by its community of users in the form of opinions. Users can collaborate by editing questions and suggesting edits to answers that have been submitted by other users. This has lead to people asking similar questions. Machine learning can help in identifying patterns in the questions and detect duplicate questions.

Quora has approached Kaggle to solve this task using their data.

Natural language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. It would be interesting to work on such a project. Hence I have chosen the Quora question pairs competition from Kaggle as my capstone project.

Problem Statement

According to Kaggle, over 100 million people visit Quora every month, and so it is normal that many people ask similarly worded questions. When multiple questions with the same intent are posted, it can cause seekers to spend more time to find the best answer to their question. It can also 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.

The task is to identify similarly worded questions and provide quality answers so that Quora writers, seekers and readers can have an improved experience.

Datasets and Inputs

The dataset(s) and/or input(s) being considered for this project is taken from Kaggle.

File descriptions
train_data.csv - the training set
test_data.csv - the test set
train_labels.csv - the training labels file
my_sample_submission.csv - provides the appropriate format for generating a valid submission
Data fields
id - the id of a training set question pair
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

Data and descriptions taken from https://www.kaggle.com/c/quora-question-pairs

Solution Statement

The goal of this project is to predict which of the provided pairs of questions contain two questions with the same meaning. This is a classification problem.
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 is taken to be 'informed' but not 100% accurate, and may include incorrect labeling. The labels on the whole dataset represent a reasonable consensus, but this may often not be true on a case by case basis for individual items in the dataset.
In machine learning, the term "ground truth" refers to the accuracy of the training set's classification for supervised learning techniques. This is used in statistical models to prove or disprove research hypotheses. The term "ground truthing" refers to the process of gathering the proper objective (provable) data for this test. Compare with gold standard. https://en.wikipedia.org/wiki/Ground_truth
I will be using training labels file for ground truthing.

Benchmark Model

Currently, Quora uses a Random Forest model to identify duplicate questions.
In this NLP project, I plan to tackle this natural language processing problem by applying 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.

I will use Random Forest model as the benchmark model and use other classification methods like Gradient boosting/ Decision trees / XG Boost/ Neural networks.

Evaluation Metrics
For this project, I will use f1 score as evaluation metric.

$$f1_{score} = 2 * \frac{precision*recall}{precision + recall}$$

$$precision = \frac{true_{positive}}{true_{(positive)} + false_{(positive)}}$$

$$recall = \frac{true_{positive}}{true_{(positive)} + false_{(negative)}}$$

where

$true_{(positive)}$ = the number of similar pairs of sentences our algorithm has successfully identified.

$False_{(positive)}$ = the number of similar pairs that are identified as non similar.

$False_{(negative)}$ = the number of non-similar pairs that are identified as similar.

The precision measures the proportion of positive instances correctly classified out of all the ones returned by the model during evaluation.

The recall measures the proportion of positive instances correctly classified out of all the positive instances returned by the model during our evaluation.
$f1_{score}$ is a single metric that captures both precision and recall.

Project Design

For this project, I plan to use the following approach.

Data exploration: Predict the duplicate questions Questions-Words/ Sentences.

Data preprocessing: Read the data and preprocess by removing the punctuations, regular expressions w/ special characters, letter case, digits. NLP words/ entity/ relationship/ most frequent or important topics Convert word to vectors/ map the vectors to the columns and identify the duplicate / non- duplicate with question1- question2

Model application: Random Forest/ Gradient boosting/ Decision trees / XG Boost/ Neural networks

Feature construction: Construct a basic set of features that can be used later to embed our samples with. Look at standard TF-IDF encoding for each of the questions. In order to limit the computational complexity and storage requirements, only encode the top terms across all documents with TF-IDF and also look at a subsample of the data. If the subsample still has very similar label distribution, continue without taking a deeper look on how to achieve better sampling than just taking the first rows of the dataset. Create a data frame where the top 50% of rows have only question 1 and the bottom 50% have only question 2 with same ordering per halve as in the original data frame.

Split words on spaces instead of using a tokenizer.
Transform questions by TF-IDF. Take the difference of all question one and question two pairs with this. This will result in a matrix that again has the same number of rows as the sub-sampled data and one vector that describes the relationship between the two questions.

Dimensionality reduction: I will be using t-Distributed Stochastic Neighbor Embedding (t-SNE) technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. The technique can be implemented via Barnes-Hut approximations, allowing it to be applied on large real-world datasets.
First, t-SNE constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked, whilst dissimilar points have an extremely small probability of being picked. Second, t-SNE defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the Kullback–Leibler divergence between the two distributions with respect to the locations of the points in the map.
Use t-SNE to embed the TF-IDF vectors in three dimensions and create an interactive scatter plot with them. Use the t-SNE model to look up the most similar words from any given point.

Evaluating Model Performance: Use cosine similarity between two vectors (or two documents on the Vector Space) to generate a metric that says how related two documents are, by looking at the angle instead of magnitude.

Example: Cosine Similarity cos(d1, d2) = (d1 · d2) / ||d1|| ||d2|| , where · indicates vector dot product, ||d|: the length of vector d
Ex: Find the similarity between documents 1 and 2.
d1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
d2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)
d1 · d2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25
||d1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)0.5 = (42) 0.5 = 6.481
||d2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)0.5 = (17) 0.5 = 4.12
cos(d1, d2) = 0.94