Identifying Quora question pairs having the same intent

Shashi Shankar
shashank@indiana.edu

Aniket Shenoy
ashenoy@iu.edu

Abstract

This paper presents a system which uses a combination of multiple text similarity measures of varying complexities to classify Quora question pairs as duplicate or different. The solution uses a support vector classifier model trained using the precomputed features ranging from longest common sub-string and sub sequences to word similarity based on lexical and semantic resources. The scope of this project is to tackle the short text similarity classification problem by applying Natural Language Processing techniques. The approach and methodologies used in this paper can be further extended to implement automatic short answer grading systems, essay grading system and textual entailment detection problems as well.

1 Introduction

Quora is a social media platform where people ask questions and can connect to the actual experts who contribute unique insights and quality answers. Quora has an enormous user base and 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 readers 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.

The problem this paper addresses is: Given a pair of Quora questions, identify whether they have the same intent i.e. is one question a duplicate of the other. In order to tackle this problem we apply Natural Language Processing techniques coupled with Machine Learning to classify a given question pair as duplicate or not. Currently, Quora uses Random Forest technique to identify duplicate questions.

Some of the research questions this paper tries to cover include: what features for text similarity would be strong predictors for this classification problem, how to judge similarity when contextual information is involved in the question pairs (i.e semantic similarity).

2 Related Work

What makes a question be a ”duplicate” of another question? Classifying questions as duplicate can be quite subjective because the true meaning of a sentence is very difficult to be known with certainty. In this dataset, labels have been provided by Human Experts. But it is prone to error as 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. It is believed that 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.

There have been many text similarity measures proposed by the researchers in the past based on surface level and semantic resources such as Gabrilovich and Markovitch, 2007; Mihalcea et al., 2006; Greedy String Tiling Wise, 1996. Classifying short texts as duplicate is similar to the problem of record linkage, deduplication etc. Databases often have same records and field values which are not syntactically identical but refer to the same entity. This is known as record linkage and it doesn't let data mining algorithms work efficiently. (Torsten Zesch et al., 2012)

Prior work in the field of short text similarity include using Deep Learning for Answer Sentence Selection, adaptive duplicate detection using learnable string similarity measures and textual entailment. Answer sentence selection is the task of selecting a sentence that contains the information required to answer a given question from a set of candidates obtained via some information extraction system (Lei Yu et al., 2014). For using learnable string similarity measures (Eneko Agirre et al., 2012) propose two learnable text distance functions for each database field according to the field's domain: an extended variant of learnable string edit distance, and a novel vector-space based measure that employs a Support Vector Machine (SVM). Textual entailment in NLP is a directional relationship between text fragments. It holds whenever the truth of one text fragment follows from another text. One problem with comparing text fragments as bags of words in vector space is that it perform sub-optimally when the texts to be compared share few words, for instance, when the texts use synonyms to convey similar messages. In the recent past researchers have started using neural networks for question answering. One widely used method is a type of siamese network for learning to project question and answer pairs into a joint space. (Lei Yu et al., 2014)

Some other popular techniques are syntactic matching of parse trees using the generative model which syntactically transforms the answers to questions. Another widely used approach is to use discriminate models over features produced from minimal edit distances between dependency parse trees. But, the problem of these approaches are that they require significant amount of feature engineering and require expensive semantic resources.

3 Data

The data is hosted by Kaggle. It consists of about 404351 training examples having the following format:

- id - the id question pair
- qid1, qid2 - unique ids of each question
- 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.

The ground truth is that The labels of whether questions are duplicate have been supplied by human experts. Thus, the labelling is prone to noise and is inherently subjective.

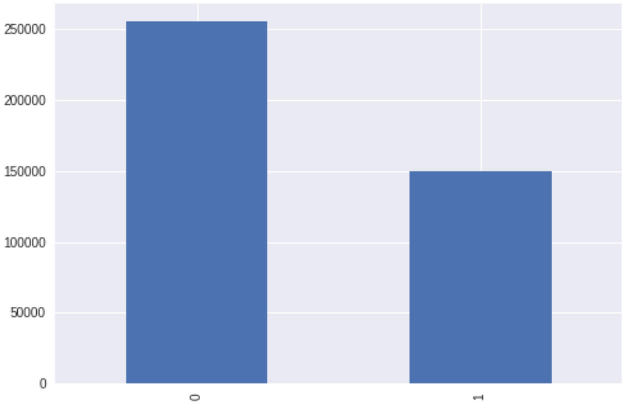

Figure 1: Label distribution in data

Here are some examples of duplicate and non duplicate questions from the data set.
The question pairs below are duplicates:

- Why did Trump win the Presidency?
- How did Donald Trump win the 2016 Presidential Election?

The question pairs below are non-duplicates:

- How can I start an online shopping (e-commerce) website?
- Which web technology is best suitable for building a big E-Commerce website?

4 Methods

Figure 3 shows the workflow of the model. The dataset is divided into a training and test set using a 70:30 split. A supervised approach is used to train the model. Being a feature based approach, the next step is to extract the features from the data.

4.1 Features:

The following features are extracted from the data (Torsten Zesch et al., 2012):

4.1.1 Simple string based measures:

- Length of questions
- Difference in length: The absolute difference in the lengths of the 2 questions.
- Number of unique characters in questions

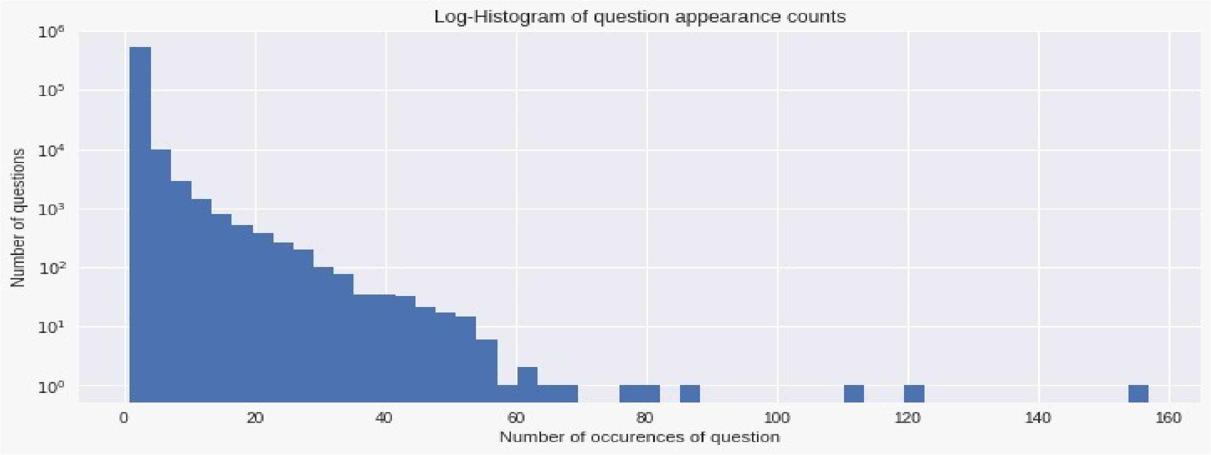

Figure 2: Histogram of question occurrence counts

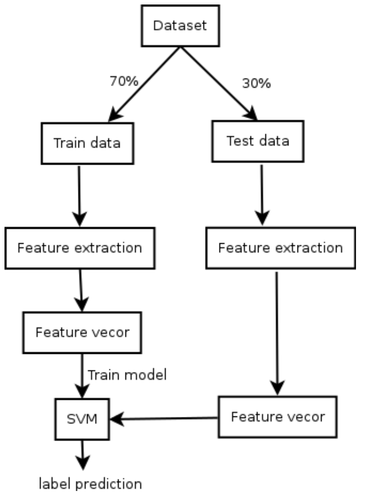

Figure 3: Model workflow

- Number of words in the questions
- Longest Common Substring: This compares the length of the longest contiguous sequence of characters (Gusfield, 1997).
- Longest Common Subsequence: This measure (Allison and Dix, 1986) drops the contiguity requirement and allows to detect similarity in case of word insertions/deletions.
- Common words: The number of common words between the two questions.

4.1.2 Stylistic and structural measure:

- Type-Token Ratio (TTR): TTR captures the statistical properties of the text. It is the ratio obtained by dividing the types (the total number of different words) occurring in a text or utterance by its tokens (the total number of words). A high TTR indicates a high degree of lexical variation while a low TTR indicates the opposite.

4.1.3 Semantic similarity measures:

Before computing the semantic similarity measures, the questions are stemmed and converted to lower case.

- Exact match: This measure calculates the number of keywords (i.e. words other than stop words) of one question that exactly match with keywords in the second question.
- Corrected match: This measure returns the number of keywords in question one that exactly match with the spell corrected keywords in the second question.
- Synonym match: This measure returns the number of keywords in question one that match with synonyms in question two.

4.1.4 Fuzzy string matching features:

Some fuzzy string matching measures such as Q Ratio (Quick ratio comparison between two strings), W Ratio (measure of the sequences' similarity between 0 and 100 using different fuzzy matching algorithms), partial token sort ratio, partial token set ratio, token set ratio, token sort ratio.

We take the log the log transform value of the features to train the SVM.

4.2 SVM:

Once the features are extracted from the data, the feature vectors are fed to the classifier for training. SVM is used as the classifier. Given a set of labelled training examples, each marked as duplicate or not, an SVM builds a binary classifier that assigns new unseen data to either one of the classes. SVMs model the input training data as points in space and try to find a separating hyperplane. Unseen test data is then mapped into the same space and predicted to belong to a category based on which side of the hyperplane they belong to. This prediction is based on the dot product of the data and the optimal weight vector that was learned during training.

The model was implemented in Python using sklearn's SVC. Other tools that were used include nltk, tqdm and fuzzywuzzy.

5 Evaluation

The evaluation metric used is the score, which is nothing but the mean accuracy on the given test set and labels. The predictions made by the trained model on the test split are compared with the corresponding labels and the mean accuracy is computed.

5.1 Results

The system proposed in the paper got an accuracy score of 85% when trained on a subset of 35000 question pairs sampled from the training data. It achieved an accuracy higher than the baseline.

6 Discussion and Conclusion

The following research question were addressed: In the feature based approach, which feature would be a strong predictor in our classifier? We did an iterative feature selection and cross validated it using train_test_split method from scikit-learn package. How various ML models perform and compare to each other for this dataset? During the implementation, SVM is compared with Naive Bayes and logistic regression. SVM outperformed the other two models. If we strip off characters which are not in the character range [a-zA-Z0-9], Will the models perform better? For questions which involve non alphanumeric characters, stripping them gives low efficiency as they lose meaning.

This paper presented a support vector classifier using string based surface similarity measures and semantic measures. It also demonstrated how effective feature selection, pre-processing and post processing are done to enhance the performance of the classifier. For future work, the model will be trained on entire training data and some more complex features such as statistical machine translation will be added to remove lexeme gaps from the input sentences. We would also like to explore how deep neural nets perform on the dataset. The features and methods used in this paper can be further extended to implement automatic short answer grading systems, essay grading system and textual entailment detection problems as well.

References

Torsten Zesch et al. 2012. *UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures*

Lei Yu et al. 2014. *Deep Learning for Answer Sentence Selection*

Yoshua Bengio, Rejean Ducharme, and Pascal Vincent 2000. *A neural probabilistic language model.*

Mikhail Bilenko et al. 2003. *Adaptive Duplicate Detection Using Learnable String Similarity Measures*

Eneko Agirre et al. 2012. *SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity*

`https://en.wikipedia.org/wiki/Quora`

`https://www.kaggle.com/c/quora-question-pairs`

`https://en.wikipedia.org/wiki/Support\_vector\_machine`

`https://en.wikipedia.org/wiki/Textual\_entailment`

`https://www.linkedin.com/pulse/duplicate-quora-question-abhishek-thakur`