SENTENCE TO SENTENCE SEMANTIC SIMILARITY

Mohammed Bilal Ansari
Faculty of Computer Science
National Research University,
Higher School of Economics
Moscow, Russia
mansari@edu.hse.ru

Abstract: There are many methods available to compare words based on the context and the meaning which convert the word into a representation in an n-dimensional vector space which is referred to as word embedding. Some of the methods are word2vec invented by Tomas Mikolov at Google, Glove from Stanford, and fastTest from Facebook. But, extending the notion of word similarity to complete sentence semantic similarity is one of the toughest problems in the Natural Language Processing(NLP). In this project, we build the system to compute the semantic similarity between two english sentences/questions using the Support Vector Machine(SVM). We consider the multiple similarities like literal similarity, shallow syntactic similarity, and latent semantic similarity to predict the semantic similarity between two sentences. Our system predicts whether two sentences are duplicate or not based on the intent of the sentences.

Keywords: sentence semantic similarity, support vector machine, literal similarity, shallow syntactic similarity, and latent semantic similarity.

I. Introduction

In Natural language processing, semantic similarity plays a vital role in many NLP applications such as Q-A chat bot, text summarization and many more. Every month over 100 million people visit Quora, 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.

In this project, we build the system, to detect the duplicate questions which have the same intent and semantic similarity. We considered the multiple similarities like literal similarity, shallow syntactic similarity, and latent semantic similarity as explanatory variables to predict the semantic similarity between two questions. This system is specially trained and built to predict the duplicate question on quora platform.

Currently, Quora uses a Random Forest model to identify duplicate questions. In this competition, in this project we proposed an SVM based system to classify whether question pairs are duplicates or not to improve the experience for Quora writers, seekers, and readers.

II. Dataset

We used the Quora Question Pairs for training and testing of our model. In this dataset, the ground truth is the set of labels that have been supplied by human experts. It consists of the following attributes or fields in the dataset.

- 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.

III. Data Preprocessing and Feature engineering

Consider the training set of the data to generate features based on similarities between the two sentences/questions. We generated the four different types of features like literal similarity, shallow syntactic similarity, latent semantic similarity

1. Literal Similarity

Literal similarity is character based similarity, which takes the characters of the both sentences into account to compute the similarities between them. Let's us have look on example,

Q1: What is the best algorithm for classification in scikit-learn?

Q2: What is the best algorithm for classification in machine learning?

Q3: Which algorithm do we use for image classification?

As we can see, question pairs(Q1, Q2) are more similar and closer in semantics than question pairs(Q1, Q3). We decided to use the edit distance over characters to computer the question-pair similarity. If the edit distance is higher for question-pair that means similarity is less and vica-versa. It might also give the worst result for similarity in case 'Do you like it?' and 'Do you link it?'. But, we got overall high performance of the system.

2. Shallow Syntactic Similarity

Question-pairt might have one or two different syntactic constituents and have very similar syntactic structures. Let's us have look on example,

Q1: How do we evaluate the model performance?

Q2: How do we assess the model performance?

As we can see, question-pair only have different 'predicates' other than that everything is the same. That means two questions may express the exact same meaning but use different sets of words as synonyms. We consider the Jaccard Similarity on a set of pos-tag to compute the syntactic similarity between questions.

$$Jaccard\_Sim(Q1,\ Q2)\ =\frac{|Q1 \cap Q2|}{|Q1 \cup Q2|}$$

where Q1 and Q2 are set of pos-tag of question1 and question2.

3. Cosine Similarity

Considering the cosine similarity to preserve more semantic information about the pair of questions, we decided to use the word2vec to build vector representation of questions and also Latent semantic analysis(LSA) model.

Vector Representation,V1 of Question, Q1:

$$V1\ =\frac{\sum_{w} TF\_IDF[w].V_w}{\sum_{w} TF\_IDF[w]}$$

$$Cosine\_Sim(V1,\ V2)\ =\frac{V1.V2}{|V1|.|V2|}$$

where V1 and V2 are vector representations of question1 and question2.

4. Normalized Word Share

We computed the normalized word share for question-pair applying jaccard similarity on collections of words of both sentences.

IV. Methodology

We decided to use the Support Vector Machine(SVM) for our binary classification problem i.e. predict the duplicate questions based on semantic similarity. Hyper Parameters for SVM has been tuned and set as follows: kernel as radial 'rbf', penalty parameter C to 0.1 (it controls the overfitting), cache_size to 5000MB (LRU cache in GPU memory to store kernel matrix values to speedup the computation).

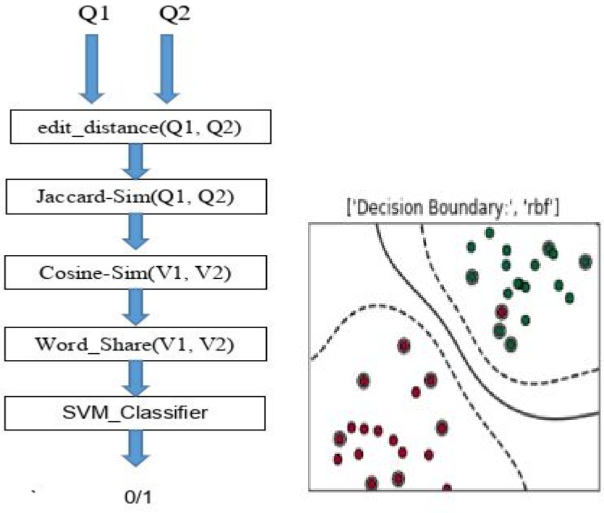

Fig: SVM for binary classification

Our corpus size is too large, training the model on scikit-learn SVM is too time taken. So, we decided to use the SVM from Raipids cuML which provides the GPU support and performs fast support vector machine classification.

V. Experiments and Results

We splitted the data set into training set and test set and considered 90% for training the system or model and 10% for testing the model. We evaluated our system of semantic similarity on two different performance metrics. We got 70.89% accuracy and 0.78 ROC_AUC score on the test set.

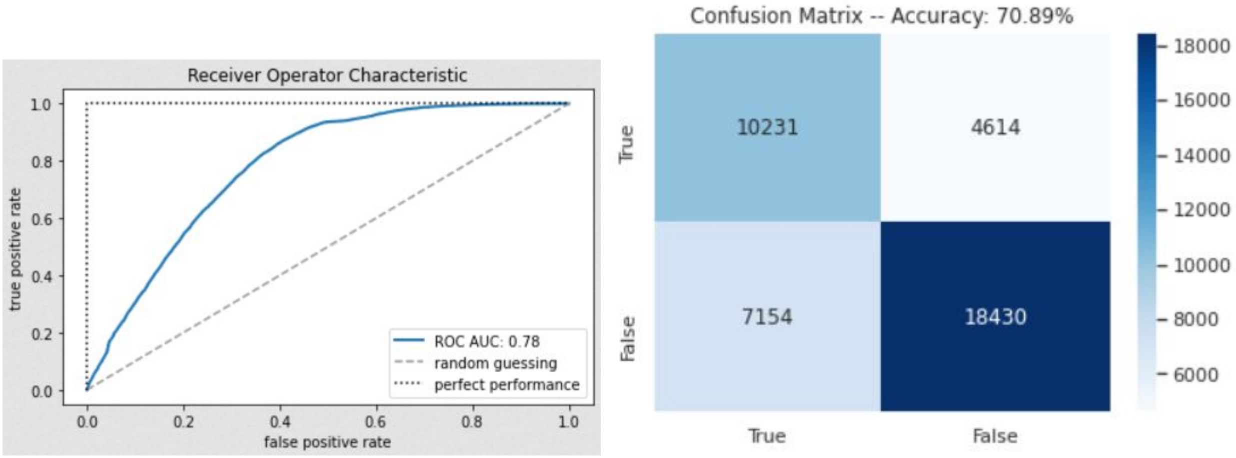

Fig. ROC_AUC Curve and Confusion Matrix

Conclusions and Future Work

Our proposed model performs well but doesn't give high performance as expected to predict duplicate questions on Quora platform. We would like to construct more sophisticated features and text pre-processing with high level of care using advanced classification techniques like XGBoost, Siamese neural network etc. to build the more robust sentence semantic similarity system in our further work.