Natural Language Understanding with the Quora Question Pairs Dataset

MID-EVALUATION REPORT

COURSE CODE: STATISTICAL METHODS IN AI - CS7.403.M21

Advisor:
Prof. Anoop Namboodiri

Team Name:
Abraca-data

Project Mentor:
Adithya Jain

Project Representatives:
Aditya Kumar Singh - 2021701010
Dhaval Taunk - 2021701028
Dhruv Srivastava - 2021701021
Lakshya Khanna - 2021900004

Academic year:
2021-2022

1. Project Description

1.1. Problem Statement

The aim is to determine whether the two questions are duplicates of each other, i.e., whether they reflect the same meaning or not, using the Quora question pair dataset. As a result, this task is essentially a binary classification problem, with a 0/1 response dependent on whether or not the questions are comparable. This project is partially a replication of the cited paper [1].

1.2. Sections Overview

The following sections in the report outlines the progress made thus far. The project status is on-track. Major focus till now has been on replicating the data pre-processing mentioned in the reference paper, building n-gram features and training linear models (SVM and Logistic Regression) on the generated features. Extensive hyper-parameter tuning is performed on models using GridSearchCV. Model with best parameters is chosen to be used for evaluation. As per the plan, we have generated a set of features mentioned in reference paper to be used in tree-based models. These features will be used in tree-based models planned to be built in the next phase.

2. Current Progress

2.1. Data Overview:

The dataset that is presently accessible has been found to be substantially unbalanced. 255,027 (63.08 %) of the 404,290 question pairings have a negative (0) label, while 149,263 (36.92 %) have a positive (1) label. The question pairs, their corresponding ids, sample id, and the accompanying label are shown in the sample from the dataset below.

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market?	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Diamond?	What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back?	0
2	2	5	6	How can I increase the speed of my internet connection while using a VPN?	How can Internet speed be increased by hacking through DNS?	0
3	3	7	8	Why am I mentally very lonely? How can I solve it?	Find the remainder when [math]23^{24}[/math] is divided by 24,23?	0
4	4	9	10	Which one dissolve in water quikly sugar, salt, methane and carbon di oxide?	Which fish would survive in salt water?	0
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and cap rising...what does that say about me?	I'm a triple Capricorn (Sun, Moon and ascendant in Capricorn) What does this say about me?	1

Figure 1: A sample from the available dataset

The dataset is being splitted into 3 sections namely train, val and test with a ratio of 70:20:10 in the similar way as mentioned by the authors.

2.2. Feature Engineering

1. **N-gram Features**: We started by creating the uni-grams, bi-grams, and tri-grams features stated in the paper by the authors. For all three type of features, the feature vector size was kept 128. These feature vectors are created using Sklearn's CountVectorizer class.

2. **Tree based features**: For creating features for tree based models, we employed the same feature engineering methods as described by the authors. A brief overview is given below:

(a) (L) Length based: length for question 1:- l1, and question 2:- l2, difference in length:- ($l1$ - $l2$), and ratio of lengths:- $\frac{l1}{l2}$.
(b) (LC) Number of common lower-cased words: count, count / length of longest sentence.
(c) (LCXS) Number of common lower-cased words, excluding stop-words: count, count / length of longest sentence.
(d) (LW) Same last word.
(e) . (CAP) Number of common capitalized words: count, count / length of longest sentence.
(f) (PRE) Number of common prefixes, for prefixes of length 3–6: count, count / length of longest sentence.
(g) (PRE) Number of common prefixes, for prefixes of length 3–6: count, count / length of longest sentence.
(h) (M) Misc: whether questions 1, 2, and both contain "not", both contain the same digit, and number of common lower-cased words after stemming, number of common lower-cased words after stemming / length of longest sentence.

These above features create a feature vector of size 25 for all the samples.

3. Training Logistic Regression on N-gram features

3.1. Introduction

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative distribution function of logistic distribution.

3.2. Replicating Previous Work & Hyper-Parameter Tuning

According to the paper, three Logistic Regression (LR) model, namely

1. LR with Unigrams (Using Unigrams Features)
2. LR with Bigrams (Using Bigrams Features)
3. LR with Trigrams (Using Trigrams Features)

were trained each with

1. $L2$ regularization,
2. controlled by α,
3. trained with stochastic gradient descent (`SGDClassifier`) using scikit-learn's implementation (Pedregosa et al., 2011), and
4. an 'Optimal' learning rate $\eta = \dfrac{1}{\alpha(t + t_0)}$

Best parameters, as given in paper, is obtained with $\alpha = 0.00001$ **and** $n\_iter = 20$ **(i.e., iterations)** for **LR Unigram**.

To implement **LR** with **SGD** as optimizer with F_1 and *Accuracy* as our evaluation metric, we took help of `scikit-learn` library.

1. `sklearn.linear_model.SGDClassifier`
2. `sklearn.metrics.accuracy_score, f1_score`

Further, to search for *optimal* parameters we applied `GridSearch` along with 5-Fold `StratifiedCrossValidation` using following packages.

1. `sklearn.model_selection.GridSearchCV`
2. `sklearn.model_selection.StratifiedKFold`

3.3. Results (LR)

N-Gram features	Original Results		Logistic Regression (with SGD)	
	Accuracy	**F_1 Score**	**Accuracy**	**F_1 Score**
Unigrams	75.4%	63.8%	68.1%	42.6%
Bigrams	79.5%	70.6%	66.9%	42.3%
Trigrams	80.8%	71.8%	64.9%	29.5%

Table 1: Replication Results

N-Gram features	Original Results		Logistic Regression (with SGD)	
	Accuracy	**F_1 Score**	**Accuracy**	**F_1 Score**
Unigrams	75.4%	63.8%	68.9%	44.4%
Bigrams	79.5%	70.6%	66.7%	36.2%
Trigrams	80.8%	71.8%	65.1%	31.9%

Table 2: Stratified 5-fold Cross-Validation Results with Hyper-Parameter Tuning

4. Training Support Vector Machine (SVM) on N-gram features

4.1. Introduction

The "Support Vector Machine" (SVM) is a supervised machine learning technique that can solve classification and regression problems. It is, however, mostly employed to solve categorization difficulties. Each data item is plotted as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a certain coordinate in the SVM algorithm. Then we accomplish classification by locating the hyper-plane that clearly distinguishes the two classes (look at the below snapshot).

4.2. Work done

Our base intention was to replicate the results mentioned in the given research-paper and hence a SVM model with same parameters was designed and tested.
It was observed that SVM with linear decision boundary can be implemented in two ways-

1. `sklearn.svm.LinearSVC`
2. `sklearn.svm.SVC` with `kernel=linear`

The given research paper had used `LinearSVC` and we were able to replicate the results and **even improved them for each of them**.

4.3. Results (SVM)

N-Gram feature	Original Results	LinearSVC	Difference
Unigrams	64.2%	**68.73%**	↑↑ **4.53%**
Bigrams	65.1%	**66.59%**	↑↑ **1.49%**
Trigrams	65.9%	**65.06%**	↓↓ 0.84%

Table 3: SVM Results on N-grams

5. Project Status

Sl. No.	Milestone	Deliverable	Timeline	Status
1	Project Proposal Submission		7th November 2021	Done
2	Data Pre-processing	Manual FE Word Embeddings	10th November 2021	Done
3	Mid-Evaluation		17th - 20th November 2021	Done
4	Model Building	Linear Models	15th November 2021	Done
		Tree-Based Models	20th November 2021	In-progress
		DL Based Models	25th November 2021	To be done
5	Result Compilation		27th November 2021	To be done
6	Final Presentation		1st - 4th December 2021	To be done
7	Final Submission		4th December 2021	To be done

6. Future Tasks

As per the proposed plan, next immediate task is to train tree-based model i.e. Decision Tree, Random Forest and Gradient Boosting on features generated. Furthermore, we would be generating CBOW embedding features that would be used in Deep Learning based models i.e. LSTM, LSTM + Attention and Transformer based model. Results from all the models and different feature sets would be compared in the final evaluation report.

References

[1] Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. Natural language understanding with the quora question pairs dataset. *arXiv preprint arXiv:1907.01041*, 2019.