Classification of Insincere Questions with ML and Neural Approaches

Vandan Mujadia, Pruthwik Mishra, Dipti Misra Sharma

MT & NLP Lab, LTRC, IIIT-Hyderabad
{vandan.mu, pruthwik.mishra}@research.iiit.ac.in, dipti@iiit.ac.in

Abstract. CIQ or Classification of Insincere Question task in FIRE 2019 focuses on differentiating proper information seeking questions from different kinds of insincere questions. As a part of this task, we (team A3-108) submitted different machine learning and neural network based models. Our best performing model which was an ensemble model of gradient boosting, random forest and 3-nearest neighbor classifiers with majority voting. This model could correctly classify 62.37% of the questions and we secured third position in the task.

Keywords: Machine Learning · Neural Networks · Adaboost · LSTM

1 Introduction

In recent years, community question answering forums have seen an upswing. The number of users of such forums has recorded exponential growth. Different toxic, malicious, hate related posts throw the biggest challenges to most of them. In this task, an attempt has been made to filter out malicious content from the forum of Quora [1] that will keep their platform more secured for users.

2 Corpus Details

The task aimed at distinguishing true information seeking questions (ISQ) from non-information seeking questions (NISQ). Six fine grained classes were designed for this classification and distribution of them in the given training corpus is shown in table 1. This task is motivated by an earlier task [2] which focused on the binary classification of sincere questions from the insincere ones. The current task is a finer counterpart of question classification posted at Quora. As the statistics of the below table suggests, the dataset is a highly imbalanced one where 2 classes constitute majority of the samples.

[1] https://www.quora.com/
[2] https://www.kaggle.com/c/quora-insincere-questions-classification/overview

Label	Description	#Samples
0	Not an insincere question	21
1	Rhetorical	488
2	Sexual Content	98
3	Hate Speech	216
4	Hypothetical	38
5	Other	38
	Total	**899**

Table 1. Label Distribution In Training Data

3 Approach

We employed two kinds of approaches for this task.

- Machine Learning Techniques
- Neural Network Approaches

3.1 Prepossessing

Preprocessing plays a vital role in tasks where the input data is in textual format. We did not use any external tokenizer for tokenizing the input. The punctuations were discarded and the white space acted as a delimiter between the words.

3.2 Feature Engineering

We used TF-IDF vectors at character and word levels for this task. We experimented with classifiers individually as well as their ensembles. Different voting procedures were also tried out. In hard voting, the class labels are predicted based on majority voting among the participating classifiers. In the case of soft voting, the voting classifier picks out the maximum of the sums of the predicted probabilities computed for the constituent classifiers. The following were implemented using scikit-learn [6] machine learning library.

- Linear SVM
- Multinomial Naive Bayes (mNB)
- Adaboost (Adaptive Boosting)
- Gradient Boost (GB)
- Random Forest (RF)
- k-Nearest Neighbor (k-NN)
- Voting Classifier

We tried various combinations of word and character level n-grams for the classification. By performing grid-search, we observed that combining both word unigrams and bigrams outperformed character level n-gram TF-IDF vectors as well as the combination of character and word level n-grams. The final submission was a hard voting classifier consisting of gradient boosting, random forest and 3-nearest neighbors classifiers.

3.3 Neural Network Models

We have also experimented with neural network based sequential classifiers, where we utilized word level features as inputs to the LSTM [3] layer (64 units) followed by Embedding layer (100 dimensions) using sequential pipeline of keras [3]. In this pipeline, we use dense output layer with softmax activation and categorical_crossentropy as loss function along with the Adam optimizer [4]. We trained this classifier for 20 epochs with an early stopping criteria. Apart from the above classifier, we have also tried combination of CNN+LSTM classifier and pre-trained glove [7] embedding+LSTM classifiers. The performance of these two classifiers were considerably poor. Therefore, we ignored them from further experimentation and reporting. In result section, we show and discuss results in detail.

4 Results

Different classifiers were trained to predict the class of each question. We include the top performing system outputs in table 2.

Model	Features	Accuracy(%)
Gradient Boost(GB)		65.35
3-NN		58.41
Random Forest(RF)		63.37
GB + 3-NN + RF(hard voting)		**62.37**
SVM	word uni + bi	61.38
multinomial NB(mNB)		57.42
Adaboost(AB)		**66.33**
GB + RF + AB(hard voting)		64.35
GB + RF + AB(soft voting)		65.34
LSTM	Words	48.51

Table 2. Accuracy of Models on Test Data

5 Observations

We could observe that boosting methods Gradient boosting and Adaboost [2] perform better than others for this task with the latter being the best. This is due to the weighted combination of different weak classifiers in Adaboost. In community QA forums like Quora, the number of spelling variations are fewer compared to social media due to character constraints. So word n-gram based TF-IDF was superior to its character counterparts. Machine learning approaches

[3] https://keras.io

outperformed the neural networks. This could be due to the higher number of parameters that deep learning approaches try to learn from a very limited amount of data.

Fig. 1. LDA word clusters on Training Data

Based on above results, we try to automatically analyze training data to understand the difficultly present in the Community Question Answering task. For that, with basic tokenization and cleaning we applied LDA [1] on the training data (without label consideration) and derived 6 text clusters from it. We used Gensim toolkit [8] for this. Figure 1 shows these derived text clusters, where Topic-5 gives hint for the Sexual content class clearly. But from rest of the topics, it is difficult to infer other classes.

We also used LDA model to analyze training text by plotting them using T-SNE [5] in two dimensions. Figure 2 represents the training text and corresponding labels that we got from LDA and figure 3 shows the text representing annotated class label from the training data. Both of these representations show that classification of these text points is quite difficult as simple topic modeling does not provide any major clues for the class boundaries.

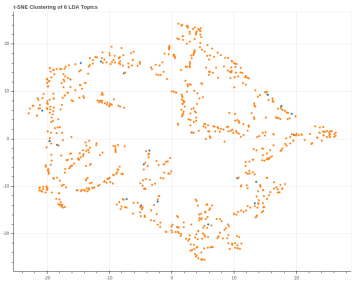

Fig. 2. T-SNE for 6 LDA Topics

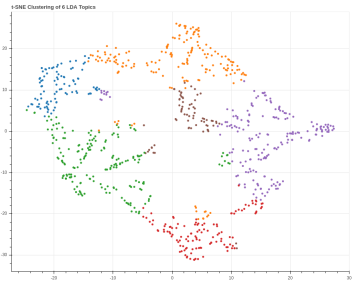

Fig. 3. T-SNE for 6 Training Topics

6 Conclusion and Future Work

We presented our supervised approaches for the FIRE task of classification of insincere questions (CIQ) in Quora for English. From our experiments, we can argue that for low resource and imbalance task such as CIQ, traditional machine learning algorithms with feature engineering outperform recent neural network based approaches. Adaboost classifier with word unigram and bigram TF-IDF features performed the best among all the classifiers. Huge amounts of unlabeled questions from Quora can be explored to improve the clustering techniques.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
2. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55**(1), 119–139 (1997)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
7. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
8. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en