Insincere Questions Classification on Quora using pre-trained word embeddings

Michael Lanier,Sibi Shanmugaraj, Nagarjuna Rao Chakka

Motivation

The primary motivation for this project is to attempt to formulate an NLP problem (predicting insincere questions) as a generalized machine learning problem and solve them using various machine learning techniques

- There are various language models that have come up such as word-2-vec, BERT etc. . . .
- Use various language models to formulate/featurize it as a machine learning problem and solve them using various generalized ML techniques.
- Benchmarks in terms of data size, algorithmic choice and hyper parameter choice.

Data Preparation

We have 1.32 million rows of dataset with unbalanced binary class labels [Nearly 6% positive example]. Neural network word embeddings were used to featurize the textual elements

- Two major categories of pre-learned word embeddings were used: word2vec and BERT
- In the word2vec embeddings, the average vector of the individual tokens is computed as the translated feature vector for the data.
- For BERT based embeddings, the average of weights of the last 4 layers of the network are used [12 layer network]. Other combinations can be used as well.
- BERT tends to provide embeddings that are much more dynamically informed

Does the data form clusters ?

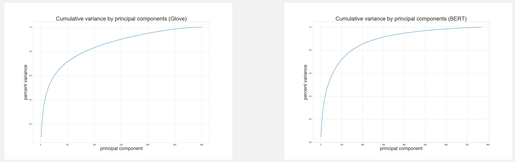
The embedded features do result in clusters that tend to isolate the binary classes (using k-means). The phenomenon was observed for both features obtained from word2vec embeddings and BERT embeddings

	GloVe		BERT	
k	precision	recall	precision	recall
4	0.143	0.00005	0.164	0.73
8	0.21	0.43	0.23	0.57
12	0.29	0.44	0.27	0.55
15	0.40	0.38	0.32	0.46
20			0.37	0.37
25			0.40	0.37

Table: Best cluster by precision and associated recall

Principal Component Analysis

For GloVe the first 100 (/300) principal components explain 82% of variance and for BERT the first 100 (/768) principal components explain 72% of the variance in the data



(a) word2vec (GloVe) (b) BERT

Figure: Cumulative explained variance

Modeling Approaches

- Logistic Regression, SVMs, Neural Networks and Ensemble Methods are tried
- Comparable Performances between Logistic Regression and Neural Networks: Neural Nets come on top slightly
- Bert embedded features have lower bias but higher variance
- Tree based ensembles didn't perform as good as Neural Networks and Logistic regression
- SVMs were computationally too expensive.

Logistic Regression and SVM

- BERT features give a better ROC-AUC perfromance, but have a slightly more variance with a reduced bias. PCA based feature transformation results in a drop in model performance

	GloVe		BERT	
Features	All	PCA 100	All	PCA 100
AUC Train	0.928	0.919	0.955	0.939
AUC Test	0.927	0.919	0.937	0.932

Table: ROC-AUC for Logistic Regression

- SVM models evaluated on precision and recall provide precision of around 0.70 and recall rate in the upwards of 0.30 (with increased regularization)
- Increasing the regularization parameter (C) increases recall. The computation time involved is huge though

Neural Networks

- Comparable performance between GloVe and BERT Embedded features. BERT tends to overfit train data.
- BERT features have lower bias but increased variance
- Increasing number of nodes overfits GloVe features; BERT model is helped by relative increase in nodes
- Comparable Performance between RELU, Leaky-RELU and Tanh activations
- 1-layer and 2-layer networks had similar performance. Adding extra nodes beyond a point didn't help.

Network	1-layer [64]	2-layer [64, 32]	2-layer [300, 100]
AUC Train	0.959	0.954	0.953
AUC Test	0.945	0.943	0.946

Table: ROC-AUC by networks design [GloVe]

- Comparable performance between GloVe and BERT embedded features on test data. BERT tends to overfit train data and needs proper regulrization.
- BERT embedded features can achieve a train data ROC-AUC of almost 1.0

Reg	0.00001	0.0001	0.0005	0.001
AUC Train	0.994	0.980	0.965	0.958
AUC Test	0.879	0.917	0.937	0.940

Table: ROC-AUC by regularization [BERT]

Random Forests [Ensembles]

- The performance of the random forests model (using Gini Coefficient) isn't quite as good as the other algorithms.
- No explicit discretization were performed.
- The model generally underfits as we see a higher ROC-AUC on the test dataset than the train dataset. Though the gap tends to reduce as we increase the number of trees.

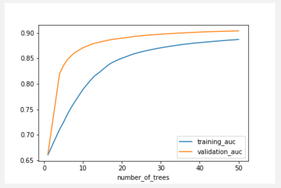


Figure: ROC-AUC for Random Forests by trees

Embedding Choice and Data Size

- Some difference in performance observed between BERT embeddings and word2vec based embeddings when used as features. BERT tend to fit well (low bias) but has higher variance (* couldn't use complete dataset on BERT due to computational constraints)
- Among word2vec based embeddings, GloVe slightly looked better though the performance among other embeddings are close.
- For word2vec based embeddings train dataset size didn't matter beyond an extent.

Conclusion

- pre-learned word embeddings serve as good features for a classification problem with text elements as features
- For this problem, Logistic regression and Neural Networks show good performance even with an unbalanced dataset. Neural Networks show the possibility of being slightly better than logistic regression when tuned.
- 2 different classes of embeddings, word2vec and BERT show variation in model performance when used as features

Future Work

- Get BERT features on entire dataset. Try other combinations of BERT featurization
- Focus on LSTM more.
- Try adding other features outside of embeddings, such as entity based features etc ...

References

[1] Xiangxin Zhu; Carl Vondrick; Charless C. Fowlkes; Deva Ramanan.
Do We Need More Training Data?
http://www.cs.columbia.edu/~vondrick/bigdata.pdf.

[2] Chris McCormick.
BERT Word Embeddings Tutorial.
https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial.