Quora Insincere Questions Classification

Srijita Ghoshal
Section A
1 November 2024
Dr. Zibo Wang

Project Information: An exploration of classical machine learning models compared to deep neural networks to see which model would classify insincere questions with the greatest accuracy

Introduction

Problem Solved

Online forum platforms have grown tremendously in size and volume. With such growth comes an increase of insincere content such as insensitive and hostile posts towards others. However, companies can't hire enough employees to regulate such volumes of inappropriate posts. Thus, it becomes a machine-learning problem to use computer models to regulate such posts. This is the exact problem Quora has launched a challenge for on Kaggle [1].

Reasoning For Choosing This Problem

The reason I decided to delve into this problem was because I wanted to explore if truly deep learning models had an advantage in solving natural language processing problems where we have a large set of words to train on. These deep learning models are often slower and much more computationally expensive, thus it would be a great benefit to get away with solving this problem with much simpler, faster, classical models.

Background Information

Natural language processing is often a challenging problem to solve due to the dimensionality and complexity of the data. For training, Quora supplied a dataset of questions posted on their website. It includes approximately 1.3 million rows in the training dataset and about 375,000 rows in the test dataset. Each question has a target classification of either 1 or 0 (insincere or not). As seen by the figure below, the average question has approximately 10 words, but some questions go up above 50 words. Thus, the models used must be able to handle such a volume of data.

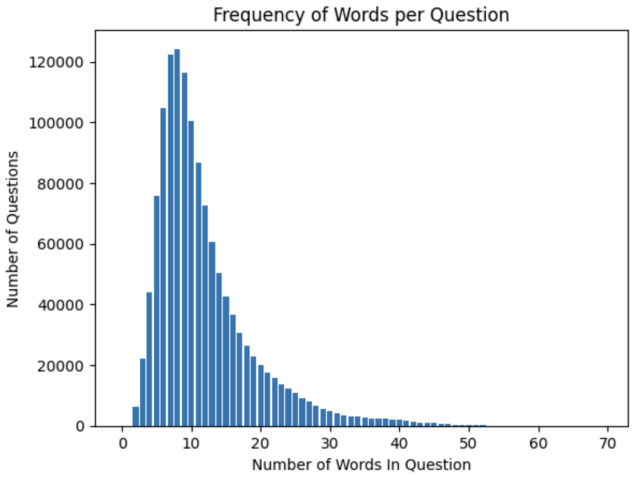

Figure 1: Frequency of Words Per Question

Definition of Success

A success model that classifies questions as insincere or not should reach closest to 95% accuracy or above. This is because it is very important to flag questions correctly. We would want to minimize false positives or false negatives where we either classify a normal question as insincere or an insincere question as normal. This would hurt the user experience. Note that this way, the machine learning model will flag the bulk of the volume, with very little human intervention if it constitutes around 95% accuracy. Thus, such a product would save lots of funds in human labor for flagging posts, which is very valuable to Quora.

Technical Details

Datasets

Quora has provided two datasets: one called train.csv, and the other test.csv. For each question (a row in the dataset), the training dataset has an identifier followed by the question content and then the classification as insincere or not. 1 in the “target” column indicates that the question is insincere. The qid is not useful in training. This translates to the model features only being the text content of the question (question_text). Note that the test data has the qid and question text but no target. Thus, we must predict the target for the test set.

Sample Training Data

Features: “question_text”	Target: “target”
Has the United States become the largest dictatorship in the world?	1

Data Wrangling & Cleaning

Given that natural language often has filler words like “the,” “and,” “or,” etc., it is important to remove these words so they do not contribute to the complexity of training. Furthermore, natural language also has punctuation that we want to ignore for training. Therefore, I went through the training.csv data and removed all filler and punctuation. These cleaned rows were placed in a cleaned_data.csv. Lastly, any questions that had no content or target were dropped from the data.

Models

Model Types

As described previously, two groups of models were compared: classical baseline models and deep neural networks. The classical baseline models used were Logistic Regression, SVM (Support Vector Machine), and Random Forest. These were compared against two deep learning models: LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network). All models had a training and test split of 80-20%. However, given the simplicity of the classical models and shorter training time, more data was used to train the classical models, and less data was used for the deep learning models.

Feature Learning

There was some feature learning applied to the classical models where Term Frequency-Inverse Document Frequency (TF-IDF) was used. This gives higher importance to words that have greater significance. The deep learning models did not have any feature learning that was preprocessed, given that these models often have their own methods of feature learning [2].

Logistic Regression

Logistic Regression is a simple machine learning model often used for binary classification. It predicts whether an input is of the target class or not. In this case, the prediction is whether the question is insincere or not [3]. The data that was fed into this model was put through a TF-IDF Vectorizer before being split into train and test data. Given this model had the lowest training time out of all models compared, I was able to use the entirety of the test data provided by Quora for training. This is approximately 1.3 million rows of questions.

Support Vector Machine

Next up, the SVC (Support Vector Classification) model was used for classification. This model is part of the SVM framework. This model draws a decision boundary between the two classes when training and later uses this decision boundary to classify new points as either one of the two classes (insincere or not) [4]. This model can often be computationally slow and consume large amounts of memory. Thus, I was not able to train this model with a huge amount of data as I was able to with logistic regression. With SVC, I trained with 100,000 rows due to memory constraints. Once again, the TF-IDF Vectorizer was used before splitting the data. Additionally, the model used a linear kernel function.

Random Forest

The last classical model was Random Forest. This model creates a number of decision trees and each decision tree will make a classification given a set of questions [5]. Decision trees can be prone to overfitting; however, the Random Forest model helps prevent this by creating multiple decision trees. For the purposes of the Quora classification problem, I trained the random forest with 100 estimators (decision trees). Furthermore, the data was put through the TF-IDF Vectorizer before being split into training and test data. For this model, I utilized 100,000 rows of data as I was experiencing performance issues with more rows.

LSTM

LSTM is the first deep learning model that was compared against the classical models. This model is a special type of Recurrent Neural Network (RNN) that uses information that was recently learned to process the next round of inputs [6]. For deep learning, tokenization is important for improving the performance of the learning. Thus, the data was tokenized before being split into training and testing sets. Given that deep learning models take lots of memory and computation power to train, only 20,000 rows were feasible for training. Furthermore, to decrease complexity, the maximum number of unique words considered were 10,000 (max_words) with a max length of these words being 100 characters (max_len). For this neural network, I utilized 6 layers. There were two Dropout layers to help prevent overfitting and the last Dense layer had an activation of softmax so the probabilities of the two classes would add to 1. The loss function used was categorical cross entropy and the model was trained for 3 epochs.

Layer	**Parameters**
Embedding	Input Dimension: max_words, Output Dimension: 128, Input Length: max_len
LSTM	128 Neurons
Dropout	Rate of 0.2
LSTM	64 Neurons
Dropout	Rate of 0.2

Dense	Activation of Softmax

CNN

CNN is the second deep learning model compared in this exploration. This model is often used for image recognition due to its strength of finding patterns in data, but it can also be used for natural language processing [7]. Furthermore, this model uses a process of convolution and pooling to learn these patterns. Similar to LSTM, data was tokenized before being split into training and testing sets, 20,000 rows were used for training, and the maximum number of unique words considered were 10,000 with a max length of these words being 100 characters. The model was trained for 3 epochs and the loss function was binary cross entropy. Again, 6 layers were used and Relu was used as the activation function to increase computational efficiency:

Layer	**Parameters**
Embedding	Input Dimension: max_words, Output Dimension: 128, Input Length: max_len
Conv1D	Filters: 64, Kernel Size: 3, Activation of Relu
MaxPooling1D	Pool Size: 2
Flatten	No parameters used
Dense	10 Neurons, Activation of Relu
Dropout	Rate of 0.5

Performance

Below is a breakdown of model performance before hyperparameter tuning was applied.

Evaluation Metrics for Classical Models

Model	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Logistic Regression	0.93	1.00	0.96	0.9315
SVM	0.94	1.00	0.97	0.9365
Random Forest	0.93	1.00	0.96	0.9325

Evaluation Metrics for Deep Learning Models

Model	**Test Loss**	**Test Accuracy**
LSTM	0.2179	0.9413

CNN	0.0958	0.9400

Model Improvement

Hyperparameter Tuning for Logistic Regression

Given that logistic regression was one of the quickest and simplest models to train, I wanted to explore the furthest capabilities of Logistic Regression on this dataset. This would serve as the best baseline for the simplest model of comparison against the deep learning models. Furthermore, Logistic Regression was the only model I was able to train with the entire dataset.

Some changes I made to the model included doing a dimensionality reduction where I kept 100 dimensions. Furthermore, I did a grid search to find the best C, the inverse of the regulation, and penalty (either L1 or L2 norm). This grid search found the best C to be **0.36** out of values between 1 x 10^{-4} and 1 x 10^{4}. The best penalty term was **L2** norm. Thus, with tuning, the final test accuracy was **0.9391** which was a slight improvement from the baseline model previously.

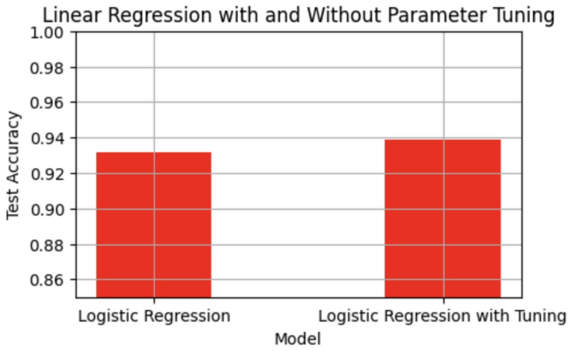

Figure 2: Linear Regression with and without Parameter Tuning

Hyperparameter Tuning for LSTM

Since LSTM had the best accuracy out of the two deep learning models, I wanted to apply parameter tuning to see how much this model could be improved and if it would be better than the results of the tuned logistic regression model. In this dataset, we want to give more weight to the insincere class of questions, given it is the minority class. Thus, I wrote a custom loss function that calculated the weighted binary cross entropy. This function gave 3 times the significance to the insincere class. Furthermore, I added a regularization term to the model using the L2 norm with a lambda of 0.1.

Along with that, I noticed the LSTM was prone to overfitting so I added an early stopping callback that would monitor the loss and restore a previous state of the model if it found signs of overfitting. Everything else in the model was kept constant to the first iteration of the unimproved model. All these changes made a decent difference to the performance of the model. The testing accuracy was **0.9455**. Although there was some overfitting due to the training accuracy being higher than the test accuracy (training accuracy of 0.9693), this was an improvement from the previous test accuracy of the model.

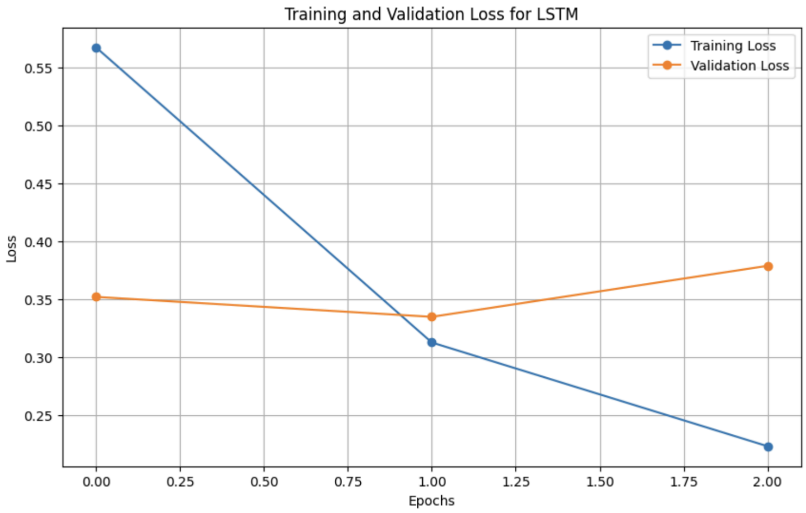

Figure 3: Training and Validation Loss for LSTM

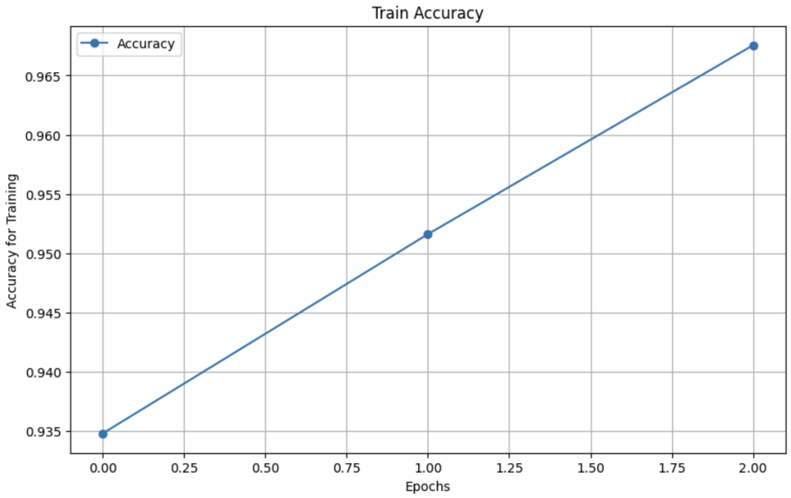

Figure 4: Training Accuracy For LSTM

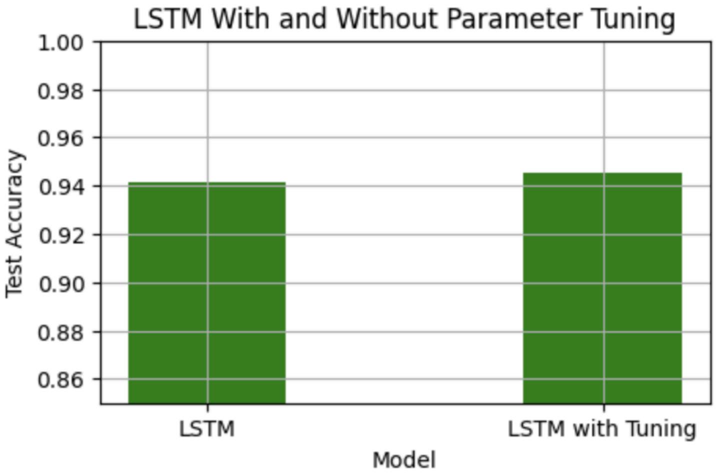

Figure 5: LSTM Accuracy with and without Parameter Tuning

Final Results

Overall, after applying hyperparameter tuning to the Logistic Regression Model and the LSTM Model, there were slight performance increases.

Model	Test Accuracy
Logistic Regression	0.9315
SVM	0.9365
Random Forest	0.9325
LSTM	0.9413
CNN	0.9400
Logistic Regression with Tuning	0.9391
LSTM with Tuning	0.9455

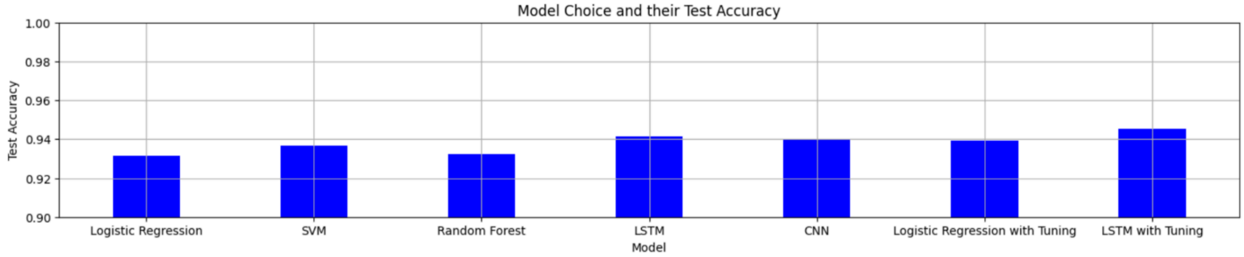

Figure 6: All Model Performances and Their Accuracy

Therefore, it seems that the LSTM Model with tuning performed the very best out of all models with an accuracy nearing almost 0.95, which is the most successful model for the Quora Insincere Classifier. It should be noted that the simple models did perform very well and that is likely due to the fact that they were easily trained with a greater volume of data. If memory and computation were not constraints, then the deep learning models could be trained with the same amount of data as the classical models and they would likely reach even higher accuracy.

Challenges and Conclusion

Challenges

One of the greatest challenges I faced during this exploration was the computational inefficiency and memory consumption of the deep learning models. I wanted to train the deep learning models with the same amount of data as the classical models, but given the constraints of Google Collab and my computer, it was not possible. Just to train the deep learning models with 10,000 rows would take 5-6 minutes. I also only trained with 3 epochs after dimensionality reduction, tokenization, and cleaning the dataset to remove unnecessary words and characters. I believe the potential for these deep learning models is much greater if I had a feasible way to train them with a greater volume of data.

Another issue I encountered was that I could not train SVM with greater than 100,000 rows and this was because SVM was taking too much memory in Collab and causing crashes. However, if memory was not a constraint, SVM would likely have a greater performance as well.

Lastly, I wanted to explore the BERT model for natural language processing, but it required an API key; therefore, I was not able to get working. However, BERT is well known for natural language processing, so I still remain curious about the possibilities of its performance with this Quora dataset.

Success Reflection

I believe I was able to achieve the desired results of 0.95 accuracy to accept the classification model as a valid product that Quora could use to remove insincere questions on their website. Although the LSTM test accuracy was just below 0.95 with an actual value of 0.9455, it was nearly there. Given that this model was only trained on 20,000 rows, this is an impressive accuracy. Since Quora has the resources to train a deep learning model with much more data, this accuracy would increase even further and scale well to larger sets of data. Thus, this LSTM model would be able to classify 95% of questions correctly and would only require 5% of questions to be checked manually. This would help tremendously in keeping their website clean.

Lessons Learned

One of the biggest lessons I learned from this project was that simple models can often work very well with large volumes of data. For example, the Logistic Regression model had a pretty good accuracy of 0.9391 with some quick hyperparameter tuning. Furthermore, given the simplicity of the model, it was much easier to train it with a large volume of data. On the other hand, the neural networks did perform better; however, the training time was magnitudes larger. The Logistic Regression took just 30 seconds, while the neural networks were around 5 minutes to train. If someone is looking for a simple, but still decently effective model, then classical models are the way to go; however, deep learning is pricey when it comes to training, but they tend to be more robust.

Future Work

Given more time for exploration, I would try to train these deep learning models with more data and on a computer with more memory and a better GPU. Thus, the true accuracy potential of these deep learning models would be revealed. I did think that training the classical models with more data was a fair trade off against the robustness of the neural networks; however, it was not a true controlled experiment given the size of the training sets were different for the two groups of models.

Additionally, I wanted to explore the potential of the BERT model given its great ability in natural language processing. In the future, I would obtain an API Key and compare this deep learning model against the classical models. I would also try hyperparameter tuning with CNN; however, I did not attempt this due to how long training was taking with just the hyperparameter tuning for the LSTM model. It would serve as an interesting comparison to see how the performance of the different deep learning models differ against each other after hyperparameter tuning.

Acknowledgment and References

Acknowledgements

I would like to give appreciation to GeeksForGeeks.org for providing in depth tutorials on how to utilize the classical machine learning models and the deep learning models. Furthermore, the website contains many explanations on what machine learning terms are and how to implement them such as TF-IDF, Vectorization, Dimensionality Reduction, etc.

SciKit Learn also had great documentation on all model parameters and how to use each model.

References

[1] "Quora Insincere Questions Classification," Kaggle, https://www.kaggle.com/competitions/quora-insincere-questions-classification/data (accessed Oct. 26, 2024).

[2] S. Bhattiprolu, "Feature learning vs.feature engineering," ZEISS arivis - The Scientific Image Analysis Platform, https://www.arivis.com/blog/feature-learning-vs-feature-engineering#:~:text=Deep%20learning%20allows%20for%20both,through%20additional%20deep%20learning%20training (accessed Oct. 26, 2024).

[3] M. Banoula, "Logistic regression in machine learning explained," Simplilearn.com, https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python#:~:text=Logistic%20regression%20machine%20learning%20is,one%20or%20more%20independent%20variables (accessed Oct. 26, 2024).

[4] "Machine Learning for Engineers," Support Vector Classifier, https://apmonitor.com/pds/index.php/Main/SupportVectorClassifier#:~:text=A%20support%20vector%20classifier%20is,be%20used%20for%20classification%20tasks (accessed Oct. 26, 2024).

[5] IBM, "What is Random Forest?," IBM, https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems (accessed Oct. 26, 2024).

[6] "What is LSTM? Introduction to long short-term memory," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (accessed Oct. 26, 2024).

[7] "What is a Convolutional Neural Network (CNN)," Arm, https://www.arm.com/glossary/convolutional-neural-network#:~:text=A%20convolutional%20neural%20network%20(CNN)%20is%20a%20type%20of%20artificial,to%20recognize%20patterns%20in%20images (accessed Oct. 26, 2024).