

**CS 605 - Smart Assistant for Natural Language Processing (G1)**

**Project Report**

Understanding and Classifying Statement Intention

Through Detecting Insincere Questions on Quora Platform

Group 4

DAVID CHONG JIAN WEI

LEE MENG YONG

NEO YI PENG

NURUL AEIYUNI HUSNA BINTI BAKRI

ZHU WANYI

**AY 2018/19 Term 3**

**16 August 2019**

Table of Contents

[Introduction 2](#_Toc16871378)

[Motivation 2](#_Toc16871379)

[Literature Review 2](#_Toc16871380)

[Process Flow 3](#_Toc16871381)

[Dataset 3](#_Toc16871382)

[Exploratory Analysis 4](#_Toc16871383)

[Tools and Resources 6](#_Toc16871384)

[Timeline 7](#_Toc16871385)

[Methodology 7](#_Toc16871386)

[1. Baseline Models 7](#_Toc16871387)

[Text Pre-processing 7](#_Toc16871388)

[Word Embedding 9](#_Toc16871389)

[Binary Classification 9](#_Toc16871390)

[2. ULMFiT (Universal Language Model Fine-Tuning for Text Classification) 10](#_Toc16871391)

[AWD-LSTM 10](#_Toc16871392)

[3. BERT (Bidirectional Encoder Representations from Transformers) 11](#_Toc16871393)

[Results and Discussion 11](#_Toc16871394)

[Conclusion and Future Work 13](#_Toc16871395)

[References 14](#_Toc16871396)

# Introduction

One of the greatest challenges existing in Natural Language Processing today is the understanding of intentions, to try to understand what the author is trying to achieve through the statement. Intentions could be pure such as asking a genuine question, or malicious such as trying to mislead, to confuse, or to aggravate readers.

This report is produced in line with the Kaggle challenge called ‘Quora Insincere Question Classification’ [1]. Quora is a question-and-answer platform that allows users to ask and answer questions to share knowledge and insights. With over 200 million monthly active users [2], Quora strives to make the platform a safe place to share and exchange information. Like other social media sites, one of the key challenges for Quora is to identify insincere questions which essentially founded upon false premises with no intention of looking for helpful answer. Currently, the platform relies on manual effort to identify such questions and content. To help with the issue, our team developed classification models which can allow Quora to detect insincere questions automatically. The team also analysed the characteristics of insincere questions to derive insights that can potentially be applied in other similar areas.

# Motivation

As a knowledge sharing platform that aims to empower users to learn from each other, the main challenge faced by Quora is to sieve out questions that have no intention of seeking fruitful discussion or genuine answers. There insincere questions tend to be malicious, toxic and misleading, which is contrary to Quora’s policy of ‘Be Nice, Be Respectful’. In order to continue to be a safe platform for users to share and grow their knowledge, it is crucial for Quora to develop scalable solutions which can detect toxic content and insincere questions effectively. Such solution helps to prevent trolls from propagating misinformation with malicious intention.

# Literature Review

A few previous works relating to topics of detecting subtle language tools such as insincerity and sarcasm treated the problem as standard classification task and attempted to identify lexical and pragmatic indicator of sarcasm or insincerity. Gonzalez-Ibanez et. al. (2011) explored the use of emoticons to identify sarcasm on Twitter [3]. Kunneman et. al. (2013) identified that intensified evaluative words and exclamations are the common markers for sarcastic text [4].

However, lexical features alone are not sufficient for detecting insincerity due to the lack of contextual guides in terms of background information of the topic. Taking an example of two questions ‘Has the United State become the largest dictatorship in the world?’, versus ‘Has the North Korea become the largest dictatorship in the world?’. In order to classify both questions accurately, the model must first understand the context [5], that United State is not a dictatorship, while North Korea is. Research by Ghosh et. al. (2017) showed that by considering conversation context, LSTM models are able to achieve better performance in detecting sarcasm in social media discussions [6].

# Process Flow

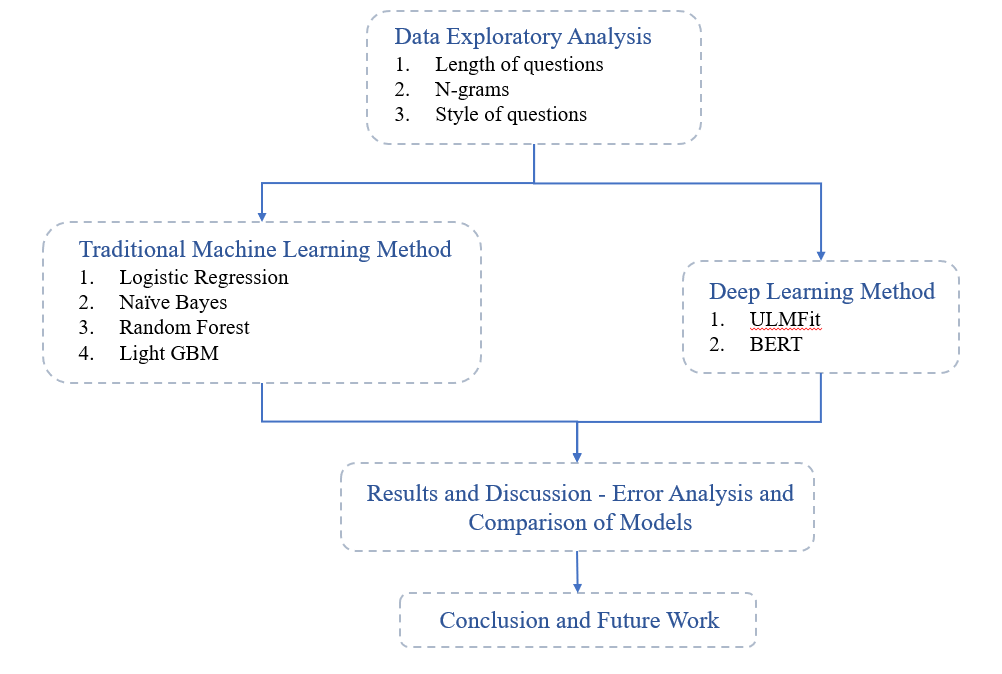


Figure Process flow diagram of the project

# Dataset

The dataset, extracted from Kaggle, contains over 1.3 million of questions that were asked on the Quora platform. In addition to question text, a target label is also given indicating if the question is insincere. For an insincere question, the target label is 1.

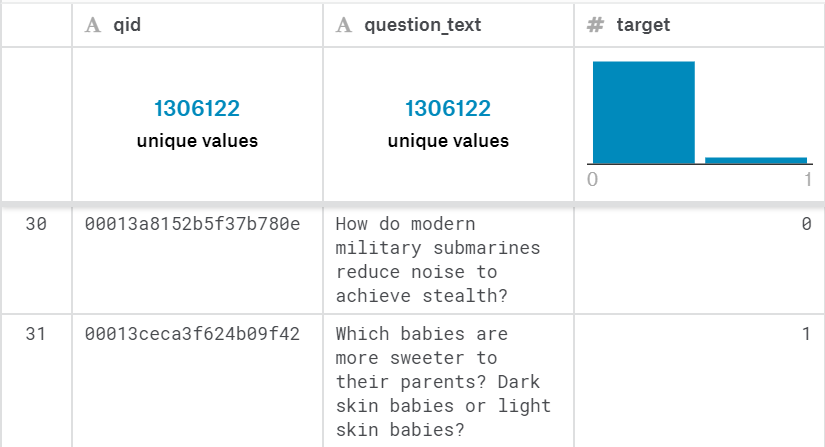


Figure Preview of training dataset

## Exploratory Analysis

Upon preliminary analysis the breakdown for each target category is as follows. 93.8% of the labelled dataset are ‘sincere’ question and 6.2% are ‘insincere’.

|  |  |
| --- | --- |
| Figure Bar chart of target class label |  |

Further exploratory is designed to look for the difference in text and writing style of sincere questions versus insincere questions. This includes the average length of questions, the n-gram for questions and the style of questions.

1. Length of each question type:

Questions are tokenized to find out the total number of words in each question. It is observed that sincere question tends to be shorter in length in comparison to insincere questions. From below box plot, the median length of sincere questions is 10 words, while the median length of insincere questions is 15 words.

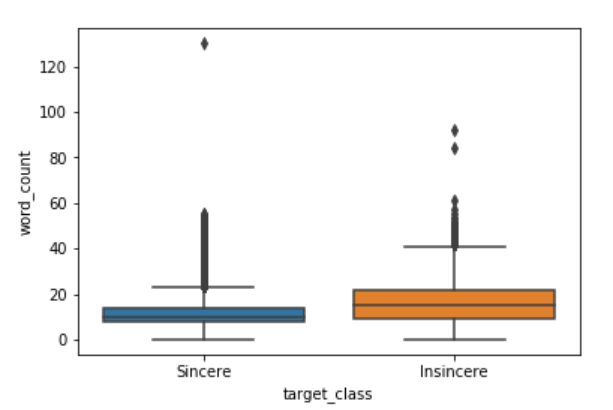


Figure Boxplot of length of sincere questions versus insincere questions

1. N-gram comparison for each question:

The objective of n-gram is to compare the common choice of vocabulary used, and the common subject for discussion for sincere and insincere questions.

**Unigram**

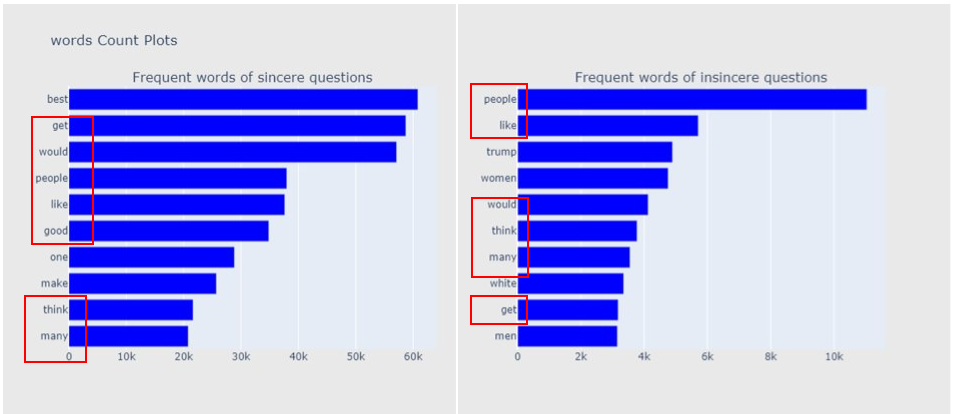
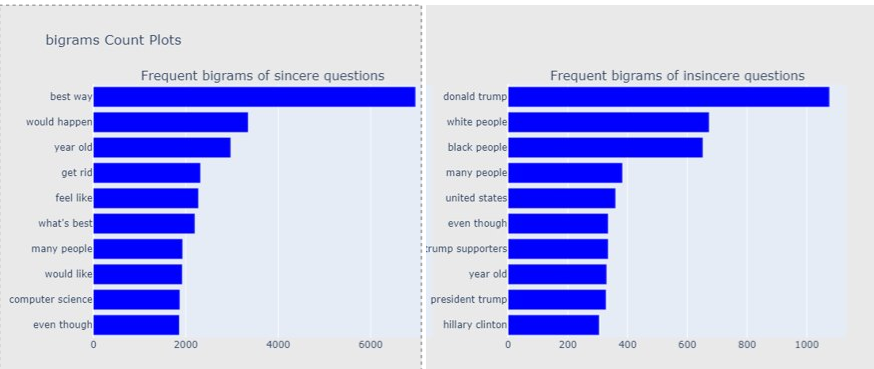




Figure Frequent unigrams of sincere and insincere questions

Comparing both side of the unigram, we observe many similar vocabularies that appears in the top 10 frequent words. Based on the unigram above, the difference observed is that in insincere questions the choice of vocabulary tends to be specific, such as ‘trump’, ‘men’, ‘women’ and ‘white’ as compared to vocabulary choice for sincere question which tend to be more generic.

**Bigram**



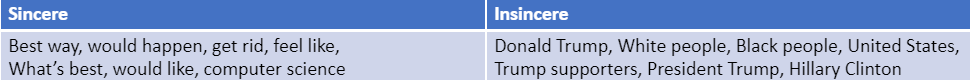


Figure Frequent bi-grams of sincere and insincere questions

We observe similar pattern in bigram. Insincere questions tend to narrow to a specific person, country, or a specific topic (e.g. race and politics). The frequent bigrams are mostly nouns. Sincere questions are more generic in vocabulary choices. The frequent bigrams tend to be hybrids of nouns, adjectives and verbs.

**Trigram**

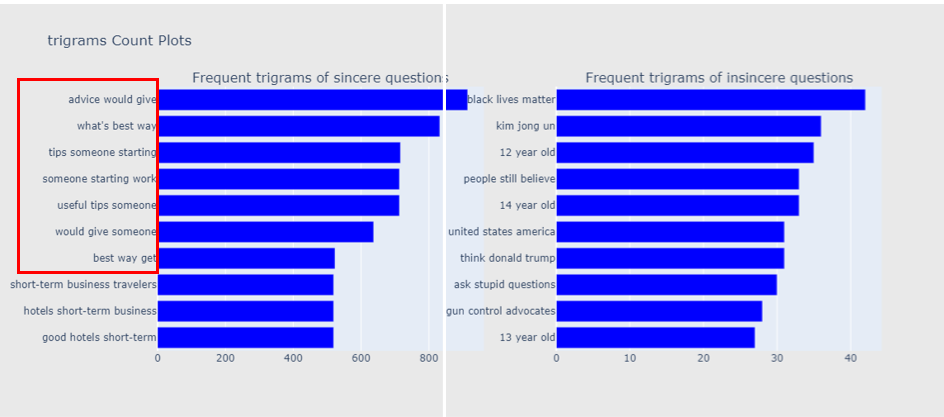


Figure Frequent tri-grams of sincere and insincere questions

From the trigram, we observe a clear distinction between the topic of interest of sincere questions and insincere questions. The phrases used in sincere question invites a discussion, while insincere questions are mostly related to some specific topics.

1. Style of questions

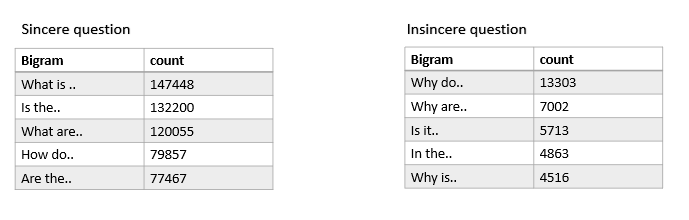


Figure Question style of sincere and insincere questions

We examine the first two words of each question to identify the style of question. Preliminary analysis suggests that insincere question tend to start the question with ‘why’ while sincere questions tend to start a discussion with ‘what’ and ‘how’.

# Tools and Resources

Tools and resources used in the project includes,

* Natural language processing packages in Python - NLTK
* Machine learning and deep learning packages - Scikit Learn, TensorFlow, PyTorch and Fastai
* Word embedding tools and datasets – word2vec (Google News vectors)
* Pre-training language representations – BERT (<https://github.com/google-research/bert>)

# Timeline

|  |  |
| --- | --- |
| Week | Tasks to perform |
| 1 July to 7 July | Exploratory data analysis and text pre-processing |
| 8 July to 14 July | Building base-line classification model with Naïve Bayes classifier, Logistics Regression etc. |
| 15 July to 21 July | Building classification model with deep learning methods |
| 22 July to 28 July | Analysing characteristics of insincere questions based on the classification model |
| 29 July to 4 Aug | Preparation of reports and presentation |

Table Project Timeline

# Methodology

## 1. Baseline Models

The baseline models are developed with traditional machine learning methods.

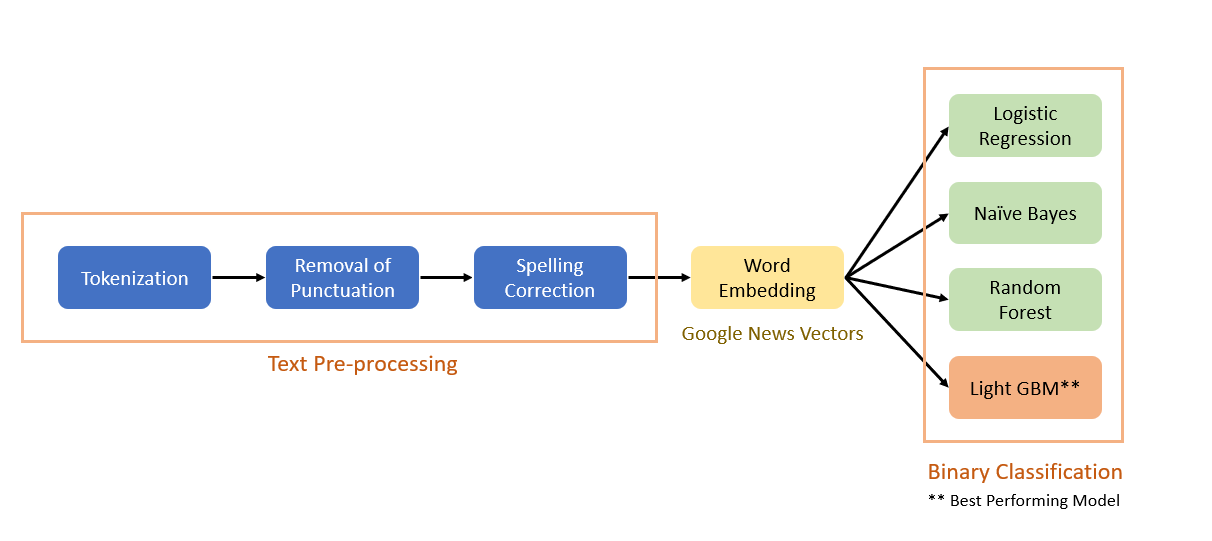


Figure Process flow of baseline models

### Text Pre-processing

Text pre-processing is an essential part of Natural Language Processing (NLP) tasks as most machine learning algorithms do not digest text directly. Commonly used pre-processing techniques for text analytics includes,

1. Tokenization – Tokenization is the process of breaking sentences into sequence of words i.e. tokens.
2. Case folding – reducing words to lowercase.
3. Removal of stop words – stop words refer to frequently occurring words that have little or no meaning, such as “the”, “that”, “is” etc. Stop words are removed because their presence can dilute the value of meaningful context related keywords.
4. Removal of punctuation
5. Stemming & Lemmatization – reducing words to its root word or base form. For example, words “working”, “works”, “worker” will all be reduced to “work”.

Not every pre-processing step listed above is mandatory and the choice of pre-processing techniques/steps is dependent on the particular text mining task. We proposed five text pre-processing workflows and determined the optimal workflow by analysing how different combinations of pre-processing steps affect the performance of classification models.

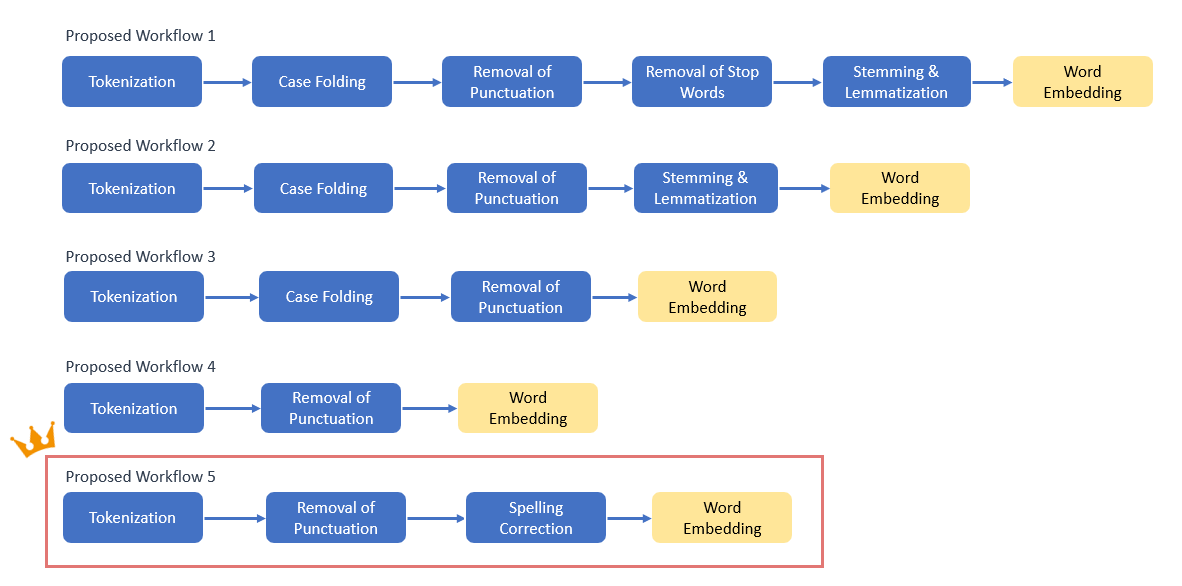


Figure Proposed text pre-processing workflows

To compare how the workflows affect the performance of classification models, Logistic regression (LR) classifier and Light GBM (LGB) classifier are used for benchmark testing. The table below summarizes the performance of classification models with different pre-processing workflow. The optimal workflow is workflow 5 with three pre-processing steps - tokenization, removal of punctuation and spelling correction.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Workflow** |  | **Accuracy** | **Precision** | **Recall** | **F1** | **Threshold-optimal F1** |
| 1 | LR | 0.939 | 0.533 | 0.177 | 0.265 | 0.447 |
| LGB | 0.948 | 0.653 | 0.353 | 0.459 | 0.550 |
| 2 | LR | 0.942 | 0.593 | 0.240 | 0.341 | 0.506 |
| LGB | 0.950 | 0.672 | 0.383 | 0.488 | 0.573 |
| 3 | LR | 0.943 | 0.598 | 0.250 | 0.353 | 0.513 |
| LGB | 0.951 | 0.679 | 0.395 | 0.500 | 0.580 |
| 4 | LR | 0.945 | 0.626 | 0.283 | 0.389 | 0.535 |
| LGB | 0.950 | 0.671 | 0.390 | 0.493 | 0.579 |
| 5 | LR | 0.945 | 0.627 | 0.289 | 0.395 | 0.538 |
| LGB | 0.951 | 0.676 | 0.400 | 0.503 | 0.584 |

Table Performance of classification model with different pre-processing workflow

We observe that having additional pre-processing steps does not always improve the model performance. Implementing case folding, stemming/lemmatization and removal of stop words is detrimental to the performance of classifier because important context related information can be lost. For example, one of the characteristics of insincere questions is the negative sentiment of question. If stop words such as “not” are removed, the overall sentiment of the question can be detected as positive instead and this affects the classifier’s capability of detecting insincerity.

### Word Embedding

After pre-processing, the question text is converted to vectors with pre-trained word embeddings model Google News Vectors. Published by Google, the model contains 300 dimensional vectors for more than 3 million words and phrases [7]. These words are in their normal form instead of stemmed form and are case sensitive (i.e. word “trump” and word “Trump” are mapped to different word vectors). By computing the mean of word vectors of each word in the question, we obtained the feature vector of the question text.

### Binary Classification

With the feature vectors, we built binary classifiers with four classification algorithms, Logistic Regression, Naive Bayes, Random Forest and Light GBM. The performance of the classifiers are presented in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1** | **Threshold-optimal F1** |
| Logistic Regression | 0.945 | 0.628 | 0.294 | 0.401 | 0.539 |
| Naive Bayes | 0.827 | 0.240 | 0.815 | 0.370 | 0.401 |
| Random Forest | 0.943 | 0.770 | 0.128 | 0.219 | 0.509 |
| Light GBM | 0.953 | 0.707 | 0.418 | 0.525 | 0.614 |

Table Performance of classifiers

The best performing model is Light GBM. After hyper-parameter tuning, we are able to achieve a F1 score of 0.614 on test set, with threshold T = 0.25. Light GBM is a gradient boosting framework based on decision tree algorithm. Different from other tree based algorithms which grow trees level-wise, Light GBM grows trees leaf-wise and thus tend to achieve lower loss [8].

## 2. ULMFiT (Universal Language Model Fine-Tuning for Text Classification)

Recurrent Neural Networks (RNNs), such as long short-term memory networks (LSTM) are effective in the realm of Natural Language Processing (NLP) or language modelling in general due to the sequential nature of text generation.

### AWD-LSTM

In general, neural networks face the problem of over-parameterization, hence it is crucial to be able to regularize the models sufficiently to enable good generalization performance. A common method of regularization is *Dropout*, where a randomly selected subset of *activations* are set to zero within each layer. This method however, is ineffective in RNNs as it disrupts the RNN’s ability to retain long-term dependencies [9]. As such, AWD-LSTM employs *DropConnect*[[1]](#footnote-2) on the hidden-to-hidden weight matrices within the LSTM, preventing overfitting from occurring on the recurrent connections of the LSTM. The dropped weights remain dropped for the entirety of the forward and backward pass.

In addition, the training algorithm is an important aspect of the model as it should aim to find a good minimum of the loss function, taking into account the speed of convergence as well. In language modelling, stochastic gradient descent (SGD) has been empirically found to outperform other algorithms such as Adam and RMSProp [9]. This model employs a variant called averaged SGD (ASGD), which differs from SGD in the solution it returns, which is an average of the iterates past a certain threshold *T*, versus the last iterate in the traditional case. The authors of the paper suggest a non-monotonic criterion to determining *T*, and proves to achieve better training outcomes. The two above-mentioned unique methods of dropout and ASGD applied in this model gives it its name - Average SGD weight-dropped LSTM (AWD-LSTM).

We ultimately used the Universal Language Model Fine-Tuning (ULM-FiT) as it employs AWD-LSTM as the language model (LM) and is able to achieve state-of-the-art results using unique transfer learning methods, including *discriminative fine-tuning* and *slanted triangular learning rates* (LR). Moreover, training is quicker as only the target task classifier required fine-tuning.

*Discriminative fine-tuning* is unique in the sense that every layer of the model uses a different learning rate (LR). Generally, the lower the number of layer, the lower the LR. *Slanted triangular LR* on the other hand, aims to allow the model to converge more quickly, hence linearly increasing the LR sharply at the start then linearly decaying based on an update schedule.

Using ULM-FiT, we achieved a F1-score of 0.69 and accuracy of 0.960, where threshold *T* = 0.3. This score is comparable to the top scores achieved in the competition. Also, it outperforms the baseline model and many traditional linear models, affirming that non-linear models, especially neural network models, are better at language modelling.

## 3. BERT (Bidirectional Encoder Representations from Transformers)

Traditional language models use RNNs, in particular Long Short-Term Memory (LSTM) networks to capture temporal sequences. One of the problems with LSTM models is that they are not parallelizable, which slows down training speed, but that is offset by its higher expressiveness [10]. However, the benefit of its expressiveness has been called into question by the Transformer [11] and convolutional models [12] beating LSTM in language modelling and translation benchmarks.

Bi-directional Encoder Representations from Transformers (BERT) [13] utilises the Transformer architecture in an encoder-decoder network to perform language modelling. A final linear layer ‘head’ can be placed on top of the encoder-decoder network to perform tasks like classification or question answering.

BERT is pre-trained on a huge corpus of English texts, including BooksCorpus and Wikipedia to make it implicitly learn the structure and subtleties of the English language. This pre-training approach allows the model to perform well in classification tasks even if the actual labelled examples are small [14] as the model already understands the language.

A key innovation of BERT is its ability to capture directional relationships in a sentence from left-to-right and right-to-left at the same time. Past models were able to capture only one direction at a time, and the best ones concatenate or average their results to mimic a bi-directional model [15]. This bi-directional approach to understanding text allows the neural network to learn relationships in both directions, just like how a human reader does naturally. BERT does so by randomly masking tokens and forcing the model to guess the masked words by using the words around it.

Using a basic BERT model without heavy engineering, we achieved a F1-score of 0.71 and accuracy of 0.963. The result matches the best F1 scores of the Kaggle competition. It was highlighted in BERT’s paper [13] that pre-trained representations reduce the need for many heavily engineered task-specific architectures. We only had to perform one epoch of fine tuning on the model to achieve the best results.

Initially, we tried using fastai’s learning rate finder to find the best learning rate for fine tuning the model. However, the model’s accuracy/F1 score deteriorated quite quickly as the learning rate was higher than the ones used in the original paper. In the end we obtained the best results using the same learning rate (3e-5) as per BERT paper [13] for fine tuning. Interestingly, more fine tuning seemed to make the model performance worse. Howard and Ruder explained that language models are prone to catastrophic forgetting which makes fine tuning a delicate task - too much can destroy the benefits of the pre-trained language model [14].

# Results and Discussion

The three classes of classification models that we have used (Light Gradient Boosting, ULMFiT, and BERT) have very different architecture and hence, different strengths in its classification ability. The motivation of the error analysis is to understand the strength of each model, and hence explore the possibility of combining the strength of each model to create an ensemble of models that perform better than any single model.

We can summarize the three models as such:

**Light Gradient Boosting (LGB)** - As LGB is trained using embedded from Google News Vectors embedding. The advantage of LGB is that it is able to correctly identify combination of word usage that are likely to result in insincere questions, such as ‘Why do you love Hitler?’, where the word ‘love’ and ‘Hitler’ appearing together usually signify insincere question.

**ULMFiT** – As ULMFiT is a version of LSTM using Recurrent Neural Network improved with transfer learning, ULMFiT considers the sequential nature of the statement, understanding the structure of the question which can indicate a rhetoric question (i.e. ‘Are pujaris of India wasting their lives?’). However, it could also misclassify sincere question with a certain sentence structure.

**BERT** - The advantage of BERT is that it is bidirectional and pre-trained on a large corpus, and hence the model can understand the context of the statement that may not be explicitly stated (such as ‘Apple is red’). However, the bi-directional structure means that it may not be as good as single directional model such as ULMFiT in understanding sentence structure and sentiments that are a key characteristic of insincere question, such as in the case of rhetoric questions (i.e. ‘Why does the media have a left-wing bias?’).

**Top questions misclassified by BERT - False Negative**

|  |  |  |  |
| --- | --- | --- | --- |
| Question | Label | Bert Prediction | Prediction Probability |
| Why do the 'Pakistani singers' are famous in India with wrost voices and wrost country they have? | 1 | 0 | 0.01 |
| If the internet is a necessity in America, shouldn't Ajit spend time in Gitmo for attacking net neutrality? | 1 | 0 | 0.01 |
| Are nationalists and right-wingers a threat to foreigners willing to immigrate to a particular country? | 1 | 0 | 0.01 |
| Do those gay porn stars have any self esteem? | 1 | 0 | 0.02 |
| Why Telugu People feel ashame to talk in Telugu in Hyderabad? | 1 | 0 | 0.02 |

Table Top questions misclassified by BERT – False Negative

The top False Negatives from BERT classifier are mostly rhetorical questions. This stems from the fact that BERT architectural is naturally bad at understanding statement structure or sentiment, while these insincere questions are easily detected by ULMFiT and LGB classifier.

**Top questions misclassified by BERT - False Positive**

|  |  |  |  |
| --- | --- | --- | --- |
| Question | Label | Bert Prediction | Prediction Probability |
| Why does my uncle touch me? | 0 | 1 | 0.90 |
| How can I chemically castrate myself without a prescription? | 0 | 1 | 0.89 |
| Should I let my 14 year old daughter sleep with my neighbor? | 0 | 1 | 0.86 |
| Was Elizabeth I sleeping with Robert Dudley's son? | 0 | 1 | 0.86 |
| What would you do if you found out that your sons are jerking off to your thoughts and have sexual thoughts about you? | 0 | 1 | 0.85 |

Table Top questions misclassified by BERT - False Positive

The top False Positives from BERT classifier appears to be the result of wrong classification from the Quora’s team, rather than the flaws of the BERT classifier.

**Model Ensemble**

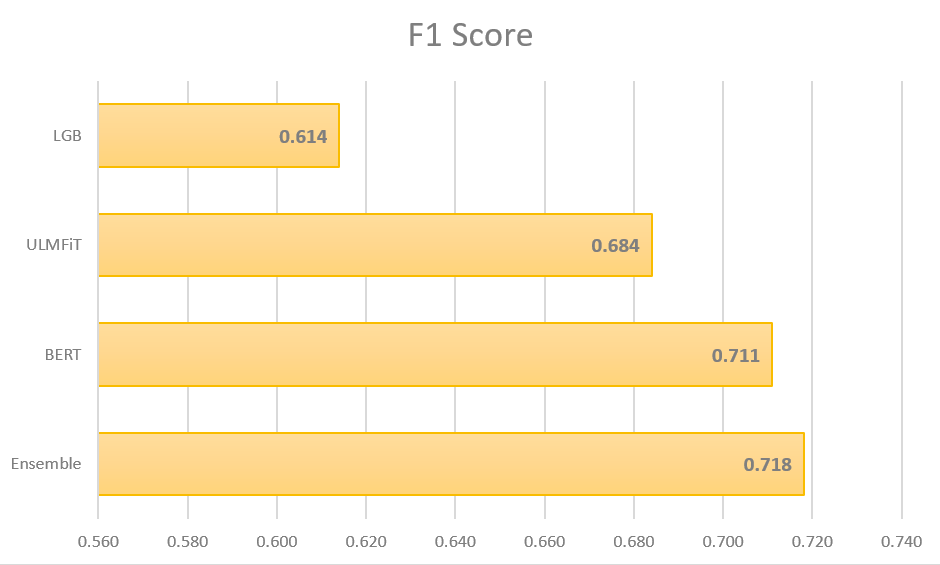
With the three models created, we are able to create a more sophisticated and accurate model by creating an ensemble of the 3 models (BERT, LGB, ULMfit). Using the logic to classify a question as insincere if at least 2 out of 3 of the models classifies a question as insincere, we are able to achieve an even superior F1 score of 0.718, which is a significant 1 percentage higher than the previous best result from solely using BERT model.

Figure 11 F1 Score of LGB, ULMFiT, BERT and Ensemble model

# Conclusion and Future Work

In this project, we used various state of the art model to identify insincere question and discussed their strengths and weaknesses. We have also proposed a new ensemble model using the three models, Light Gradient Boost, ULMFiT, and BERT to create a more accurate classifier.

Since Google published BERT in November 2018, Facebook AI Research (FAIR) have published **RoBERTa**in 29 Jul 19 [16] and Baidu Research published **ERNIE 2.0** in 31 Jul 19 [17], which outperformed BERT in a series of benchmarks. Potential future work of this project could be implementing the newly released infrastructure to further improve the performance of classifier.

Furthermore, with some fine tuning, the same models could be used in many similar context, such as identifying toxic comments, inaccurate statement, trolls, or fake news. As the internet gradually playing a bigger role in our society, our models could be employed to safeguard the internet and ensure that the internet plays a positive role in promoting correct and clean information dissemination.

# References

[1] Quora Insincere Questions Classification. (n.d.). Retrieved June 26, 2019, from <https://www.kaggle.com/c/quora-insincere-questions-classification/data>

[2] Team, T. (2018, July 03). Can High Engagement Levels Be A Competitive Edge For Quora? Retrieved June 23, 2019, from <https://www.forbes.com/sites/greatspeculations/2018/07/03/can-high-engagement-levels-be-a-competitive-edge-for-quora/#773e25715080>

[3] González-Ibánez, R., Muresan, S., & Wacholder, N. (2011, June). Identifying sarcasm in Twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2* (pp. 581-586). Association for Computational Linguistics.

[4] Kunneman, Liebrecht, Van Mulken, & Van Den Bosch. (2015). Signaling sarcasm: From hyperbole to hashtag. Information Processing and Management, 51(4), 500-509.

[5] Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., & Mihalcea, R. (2018). CASCADE: Contextual sarcasm detection in online discussion forums. *arXiv preprint arXiv:1805.06413*.

[6] Ghosh, D., Fabbri, A., & Muresan, S. (2017). The Role of Conversation Context for Sarcasm Detection in Online Interactions.

[7] Google Code Archive - Long-term storage for Google Code Project Hosting. (n.d.). Retrieved July 31, 2019, from <https://code.google.com/archive/p/word2vec/>

[8] Microsoft. (n.d.). Microsoft/LightGBM. Retrieved July 31, 2019, from <https://github.com/microsoft/LightGBM/blob/master/docs/Features.rst>

[9] Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*.

[10] Sundermeyer, M., Oparin, I., Gauvain, J. L., Freiberg, B., Schlüter, R., & Ney, H. (2013, May). Comparison of feedforward and recurrent neural network language models. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8430-8434). IEEE.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*(pp. 5998-6008).

[12] Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017, August). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1243-1252). JMLR. org.

[13] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[14] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

[15] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding with unsupervised learning*. Technical report, OpenAI.

[16] Johnson, K. (2019, July 29). Facebook AI's RoBERTa improves Google's BERT pretraining methods. Retrieved August 16, 2019, from <https://venturebeat.com/2019/07/29/facebook-ais-roberta-improves-googles-bert-pretraining-methods/>

[17] Baidu’s Optimized ERNIE Achieves State-of-the-Art Results in Natural Language Processing Tasks. (n.d.). Retrieved August 16, 2019, from <http://research.baidu.com/Blog/index-view?id=121>

1. *DropConnect* instead sets a randomly selected subset of *weights* within the network to zero. [↑](#footnote-ref-2)