



A PREDICTIVE STUDY OF SARCASTIC COMMENTS  
CONTAINED IN REDDIT FORUM COMMENTS

IOD CAPSTONE PROJECT

BY YUAN CHAN

22/08/21

## Contents

INTRODUCTION .....	3
PROJECT BUSINESS / STAKEHOLDER CASE .....	3
Business / commercial context.....	4
Social / Cultural context .....	4
Linguistic context.....	4
DATASET .....	5
PROJECT METHODOLOGY.....	6
Background.....	6
Staging .....	7
DETAILED REVIEW .....	8
Stage 1 – Data Cleaning / Wrangling.....	8
Stage 2 - Exploratory Data Analysis (EDA) .....	8
Stage 3 - Text-preprocessing .....	13
Stage 4: Feature extraction and engineering .....	14
Stage 5A.....	17
Stage 5B.....	18
BERT Encoder.....	18
Bidirectional LSTM.....	19
Stage 6 - Model Evaluation and discussion of results .....	21
Stage 5A results .....	21
Stage 5B results .....	22
Test predictions .....	23
CONCLUSIONS AND FUTURE DIRECTION.....	24
REFERENCES .....	25

## INTRODUCTION

Accurate identification of sarcasm can be challenging even for humans conversing face to face, as expressions of sarcasm are often culturally nuanced and contingent on individual perceptions of humour. Notably, the context in which a sarcastic statement is made is crucial towards its identification.

In this context, sarcasm detection by machine learning approaches is seen by many researchers as a key but challenging natural language processing (NLP) task.

Sarcasm can be expressed in various subtle ways and accurate identification requires “a deeper understanding of natural language that standard text categorization techniques cannot grasp” [1]. Adding to this complexity is the further difficulty of extracting contextual clues using conventional text classification approaches (such as “Word2Vec” and “Glove” word embeddings) as these do not account for word order or the presence of surrounding sentence features [2].

In this project, a Deep Learning (supervised learning) approach combining Transformer architecture with recurrent neural networks (RNN) layers is proposed to address the limitations of conventional machine learning approaches in classifying labelled sarcasm in text. It is anticipated that the ‘self-attention’ ability of Transformers to disambiguate word context together with the left-to-right and right-to-left “memory” of bi-directional recurrent neural network RNN layers will provide significant improvements in the identification of sarcastic comments.

The investigated project dataset consists of 150,000 comments containing sarcasm scraped from Reddit forum postings.

## PROJECT BUSINESS / STAKEHOLDER CASE

According to the Merriam Webster dictionary web-page [3], sarcasm is defined as:

*“A sharp and often satirical or ironic utterance designed to cut or give pain” or*

*“A mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual”*

Given its prevalence as an expression of subtle disapproval or even hostility, the accurate identification of sarcasm by machine learning tools is foreseeably of interest to a broad range of project stakeholders.

## Business / commercial context

From a business / commercial standpoint, a significant proportion of customer feedback is now expressed through online feedback portals, with textual interpretation and classification of customer feedback handled using machine learning tools.

Customers commonly employ sarcasm to express frustration or exasperation online. Machine learning algorithms relying solely on isolated, non-contextualised text features (such as basic sentiment polarity measures) to gauge customer satisfaction are likely to perform poorly in detecting sarcasm.

The consequences of consistently misinterpreting / miss-classifying negative sarcastic statements as positive customer feedback, such as: *"I just love how this electric shaver gives up on me every-time I'm rushing off to work"* (with the term "love" being given disproportionate significance) could result in significant degradation in customer approval and retention rates. As such, the development of machine learning approaches to detect sarcasm accurately and consistently has tangible commercial significance.

## Social / Cultural context

A significant proportion of social interaction now takes place in online forums, where the active moderation of perceived offensive or aggressive behaviour by forum posters is necessary to maintain the harmony of online communities.

Sarcasm is harder to detect by humans in an online setting where physical expressions and tonal cues are entirely absent. Un-detected sarcasm can often be a cause for misinterpretation and even offense hence sarcasm identification by computer applications is of relevance to online social moderation interest groups.

As an example, the Reddit forum invites posters to clarify if a statement being made is sarcastic through the use of a "\s" marker [4]. Whilst this facility is available, it is entirely voluntary and may not be implemented by posters.

To facilitate a more comprehensive labelling of sarcasm, a machine learning application that identifies sarcasm as part of an online forum could be useful. For instance, the application could proactively and tentatively highlight what appears to be sarcasm before inviting a poster to confirm if the apparent sarcasm was intended.

## Linguistic context

Machine learning breakthroughs in identifying sarcasm could be employed to assist the next generation of Artificial Intelligence to identify linguistically related / similar forms of complex human expressions, such as irony and humour. This will provide a significant advancement in the broader representation of human behaviour using AI technology.

## DATASET

The dataset addressed by this project is the “Sarcasm on Reddit” dataset hosted on the Kaggle website.

This dataset is “a collection of 1.3 million comments containing sarcasm from the Internet commentary website Reddit. The dataset was generated by scraping comments from Reddit containing the “\s” (sarcasm) tag. This tag is often used by Redditors to indicate that their comment is in jest and not meant to be taken seriously, and is generally a reliable indicator of sarcastic comment content.” [5].

The dataset is made available as a balanced dataset, with sarcastic and non-sarcastic comments represented equally (ie. 50-50).

This balanced ratio establishes a baseline accuracy of 50% and this must be significantly exceeded by any proposed machine learning algorithm for its predictive capacity to be non-trivial.

In this project, the investigation is constrained to a 150,000 row subset of the dataset extracted by random sampling. The sampling is stratified to retain 50-50 proportions of sarcastic versus non-sarcastic comments.

The dataset consists of 8 feature columns:

- **‘label’**: This is the target variable with **0** representing a non-sarcastic comment and **1** indicating a sarcastic comment.
- **‘comment’**: This is the posted comment and the text labelled for sarcastic content.
- **‘parent comment’**: This is the preceding comment posted and is the context to which the ‘comment’ is posted.
- **‘author’**: The online name of the poster
- **‘subreddit’**: This is the topic forum where the comment is posted
- **‘ups’** and **‘downs’**: These are up-votes and down-votes posted by readers in response to the comment made
- **‘score’**: A score is determined by the up-votes and down-votes
- **‘created UTC’**: Date and time of posted comment

label	comment	author	subreddit	score	ups	downs	created utc	parent_comment
0	The title of this article should be "How to not have sex ever again"	xNOM	MensRights	1	1	0	2014-11-07 11:48:34	You can approach women without being creepy
0	What a waste opportunity... at least be funny with it.	Jacked1218	MMA	5	5	0	2016-07-05 22:06:15	Nate Diaz Snapchat Hacked
1	But...but... sodium!	Chicup	fatlogic	1	1	0	2015-03-09 13:16:40	Canned soups have been hugely helpful for the past year's weight loss. I can have 2 cups for 160-200 calories, and feel completely full for a good while. Plus, given that I can work 12-15 hour days, I do NOT have the time to be making a fresh, home-cooked meal on a daily basis.

## PROJECT METHODOLOGY

### Background

The deep learning modelling approach in this project is inspired by a research paper “A large Annotated Corpus for Learning Natural Language Inference” by Bowman, et. al [6].

This paper explores the identification of the semantic categories of “entailment” and “contradiction” contained in pairs of sentences by a neural network based model, supported by a large corpus of 570,000 pairs of representative sentences for each of the categories [6].

A man inspects the uniform of a figure in some East Asian country.	<b>contradiction</b> C C C C C	The man is sleeping
An older and younger man smiling.	<b>neutral</b> N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	<b>contradiction</b> C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	<b>entailment</b> E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	<b>neutral</b> N N E C N	A happy woman in a fairy costume holds an umbrella.

Table 1: Randomly chosen examples from the development section of our new corpus, shown with both the selected gold labels and the full set of labels (abbreviated) from the individual annotators, including (in the first position) the label used by the initial author of the pair.

In this project, it is proposed that the classification of “entailment” and “contradiction” is a highly relevant and significant marker in sarcasm detection, given that sarcastic comments can often display elements of contradiction between the statement made and the context being responded to. It is observed that the architecture employed for a neural network identifying “contradiction” is relevant to, and may be adapted to the identification of a model to detect sarcasm.

The modelling adopted in this project is borrowed and adapted from an example Keras model framework used for the above Natural Language Inference study [7]. This model is structured to permit the side by side comparison of 2 separate sentences for “contradiction” or “entailment”.

In the case of the Reddit dataset, the use of sentence pairs combined with a deep learning approach also permits the comment text (“comment”) and the background of the comment (“parent\_comment”) to be explicitly accounted for. This permits the individual sentences to be examined and contrasted as discrete entities during training, compared to common approaches that simply concatenate the 2 sentences into a single piece of text, potentially “diluting” contextual clues.

It is further proposed that a Deep Learning model pre-trained to first identify “entailment” and “contradiction” can be fine-tuned on a new dataset to identify sarcasm. Conventional machine learning approaches rely on extracting text features unique to the dataset in question. Such approaches do not generalise well and require explicit operator intervention for feature extraction.

In summary, deploying a transfer learning approach pre-trained on an external dataset (such as for classifying contradiction), that is similar to, but still sufficiently differentiated from the dataset in question, may significantly improve the generalisation ability of predictive models.

## Staging

The project will be carried out in the following 6 stages:

- **Stage 1:** Data cleaning / wrangling – This includes checking and removal of missing values / errors and data transformation into suitable formats for processing by supervised learning.
- **Stage 2:** Exploratory Data Analysis (EDA) – EDA will be carried out on the dataset to identify salient trends / characteristics, detect anomalies, test hypotheses and check assumptions with the assistance of summary statistics and visualisation tools. Identified trends may assist in the extraction of sarcastic comment features, per stage 4 below.
- **Stage 3:** Text pre-processing – This stage involves parsing of text to remove unwanted text formatting (such as upper-casing), tokenization of text, text lemmatization, stop-word identification and part of speech tagging using Spacy / NLTK.
- **Stage 4:** Feature extraction / engineering - This stage involves extracting suitable features to detect sarcasm for conventional machine learning approaches. One category is meta-features, such as word counts, stop word counts, punctuation counts and the length of characters. Another category is word embedding vectorization – refer ‘Detailed Review’ discussion section of this report.
- **Stage 5:** Modelling – Modelling will be carried out in 2 separate phases:-

**Phase 5A** involves conventional supervised learning approaches using 5 separate classification models:

- Naive Bayes
- Logistic Regression
- Support Vector Machines (Linear SVC)
- Random Forest (Bagging ensemble)
- Gradient Boosting (Boosting ensemble)

The results obtained in Phase 5A serve as a performance benchmark to compare Deep Learning approaches.

**Phase 5B** involves Deep Learning modelling using pre-trained transformer architecture (Bidirectional Encoder Representations from Transformers (B.E.R.T) combined with recurrent neural network (Bidirectional LSTM) layers (BERT + BLSTM).

BERT + BLSTM will be compared against the following benchmarks:

- Scoring metrics from Phase 5A
- Scoring metrics for a basic feed forward multi-layer perceptron model

For both phases A and B a train-test-split of 80/20 will be adopted. 15% of the train data will be further set aside for validation for Deep Learning approaches.

- **Stage 6:** Model evaluation and prediction case – Train and test scores will be reviewed with suitable evaluation metrics such as ROC curves / learning curves. Illustrative sample predictions will be made using the adopted modelling approach.

## DETAILED REVIEW

### Stage 1 – Data Cleaning / Wrangling

#### Data Cleaning:

The dataset contained minimal rows with missing values /errors. 53 rows containing NaN values were dropped in total.

#### Wrangling:


**'UTC' date-time column** was separated into individual 'year', 'month' and 'hour' for numeric feature extraction.

**'parent\_comment'** text was reduced to a maximum of 2 sentences or 120 words. The limit of 2 sentences was imposed to minimise excessive "noise" during feature extraction. 120 words was decided on the basis that the sequence limit for the BERT encoder adopted in the Deep Learning phase is 128 words. As an additional consideration it is noted that there are only 26 words contained in 'parent\_comment' at the 75<sup>th</sup> percentile, which further validates the limits imposed above

### Stage 2 - Exploratory Data Analysis (EDA)

The following dataset features were identified in the EDA:

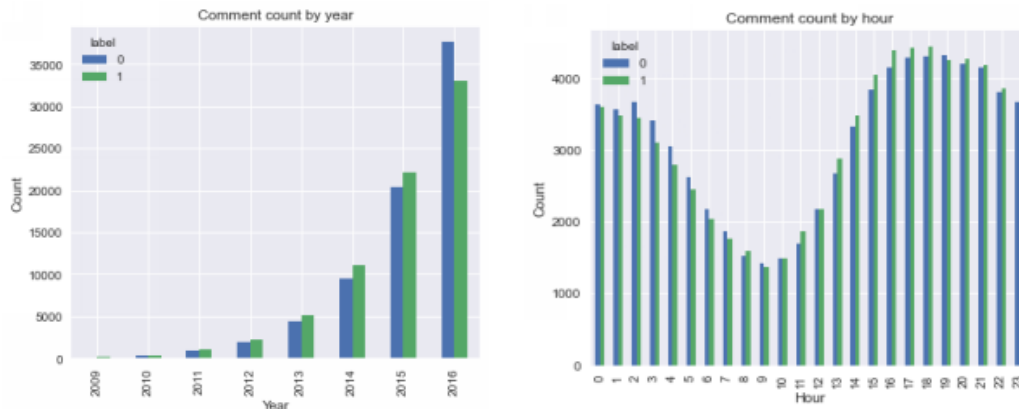
- **Self-labelling:** It is highlighted that all labelling of comments has been performed by the user and not independently verified. As such, some proportion of comments are expected to be un-labelled or labelled incorrectly. As illustration, the statement below has been made in response to a 'Black Lives Matter' posting. Although the comment is clearly sarcastic in nature, it has been labelled as 'non-sarcastic'.



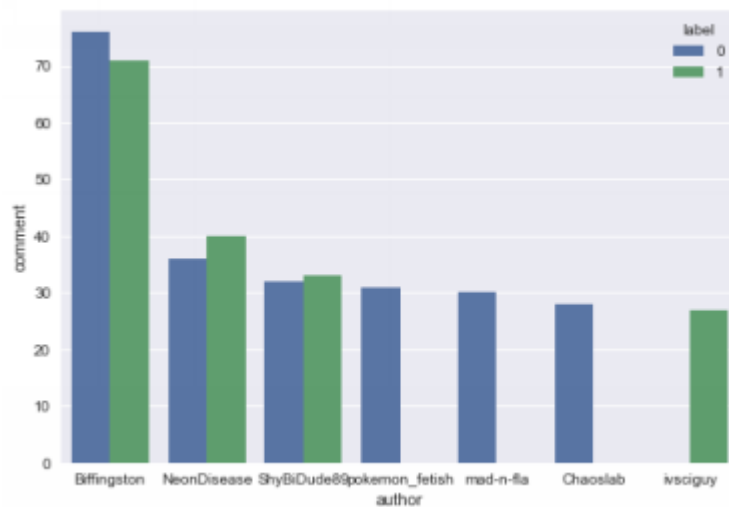
0 How about a No Lives Matter, for the incurably misanthropic?



- Label balancing:** The original dataset is highly imbalanced with a true (sarcastic) ratio of 1:100 [5]. The artificial balancing of labels to reflect 50-50 proportioning is useful from a training standpoint as larger samples of sarcastic comments improve the familiarisation of an algorithm with sarcastic features. However, the downside is that inherent trends in the dataset with respect to sarcastic posts are obscured and can no longer be inferred directly. This is evident in the graphs below showing comment posts by month, year and hour.



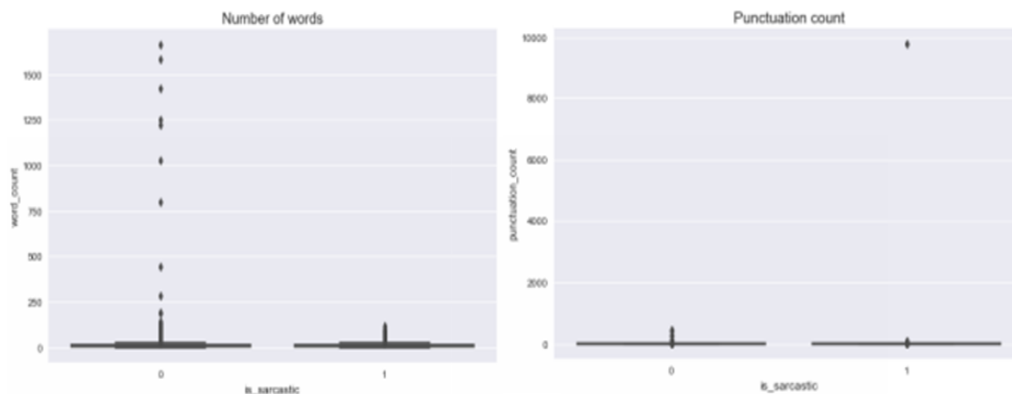
A number of 'author' records appeared similarly balanced. However, it was notable that several of the top 10 posters posted either completely sarcastic comments or none at all.



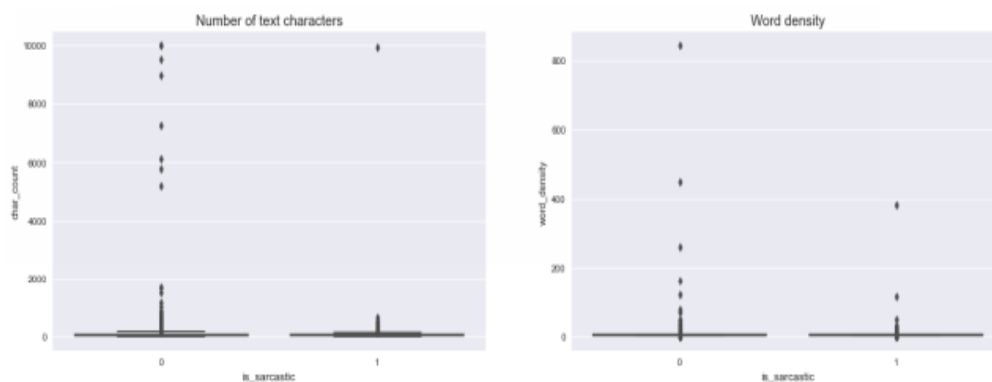
This observation suggests that the personality of individual posters may have potential as predictive features for sarcasm. Poria et. al have suggested that classification of personality from text can be performed using pre-training on a corpus labelled according to different personality traits [1]. An example of this is the Matthews and Gilliland corpus based on five personality traits of: **O**penness, **C**onscientiousness, **E**xtraversion, **A**greeableness and **N**euroticism (OCEAN) [8].

- **Length of sentences, punctuation count (Sarcastic vs Non-Sarcastic)**

Boxplots comparing text meta-features identified minimal differences between Sarcastic Vs Non-Sarcastic comments. This suggested that the meta-features have low correlation with the target variable and are relatively ineffective as predictive markers. A correlation heat-map confirms this – refer ‘Detailed Review’ section of this report for Stage 4.



- **Number of text characters / word density (Sarcastic Vs Non-Sarcastic) – Minimal differences in boxplots**



- **Top 10 most-common words**

The results for top 10 most common words in sarcastic comments unsurprisingly feature many common terms such as “yeah”, “not”, “right” and “really” that would otherwise be classified as stop words. Although some of these terms also feature in non-sarcastic comments, the difference in the number of terms showing up in either category is significant in many instances. For example, the term “yeah” features 4714 times in sarcastic posts but only 1345 times in non-sarcastic posts. This strongly suggests “yeah” as a significant marker for sarcastic comments and supports the consideration that stop words should be retained for model training and prediction.



- **Topic identification**

For the purposes of topic identification, in contrast to text preprocessing for model training, the predominance of common stop words featuring in sarcastic comments does not facilitate the identification of clearly separated topics. The difficulty of topical definition is further amplified by the diversity of topics represented in the comments, each belonging to one of over 14000 subreddit groups.

```
# Check number of subreddit groups with most number of posts
df_full.subreddit.value_counts()

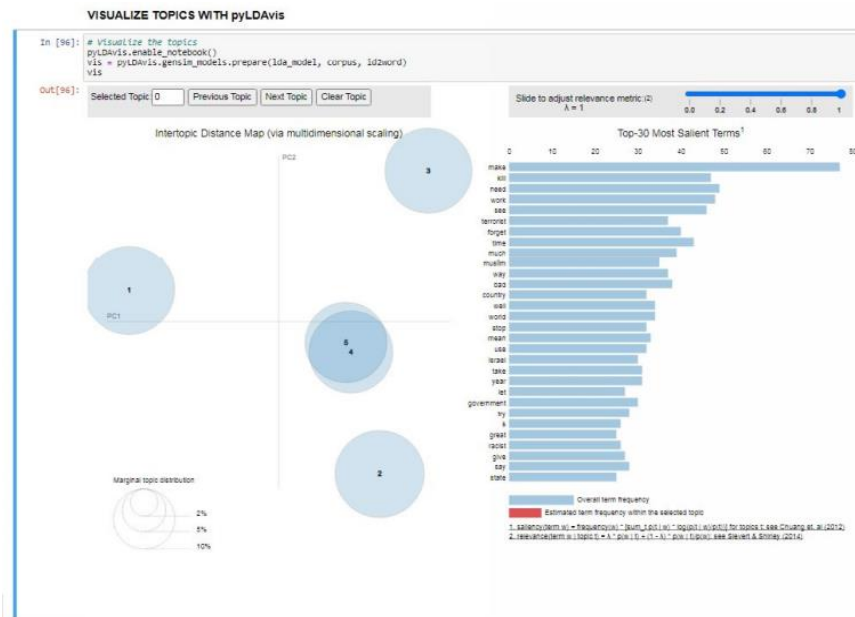
AskReddit      65674
politics       39493
worldnews     26376
leagueoflegends 21034
pcmasterrace  18987
...
hungarian      1
NFL2go         1
elixir         1
TheFalloutDiaries 1
texasbeer      1
Name: subreddit, Length: 14876, dtype: int64
```

Topic identification is practicably achievable only by focusing on a specific subreddit group.

For the “world news” subgroup, after stop word filtering is applied, Latent Dirichlet Allocation (LDA) analysis yields reasonably well defined topics addressed in sarcastic comments with clear topic separation as indicated via Word Cloud and PyLDAvis visualisation:



```
[
(0,
'0.020*need" + 0.020*work" + 0.015*way" + 0.013*mean" + 0.013*use" + '
'0.013*take" + 0.011*say" + 0.011*problem" + 0.011*american" + '
'0.010*america'),
(1,
'0.018*terrorist" + 0.017*muslim" + 0.016*country" + 0.012*great" + '
'0.011*attack" + 0.009*look" + 0.009*tell" + 0.009*police" + '
'0.008*china" + 0.008*jew'),
(2,
'0.030*make" + 0.018*see" + 0.017*time" + 0.015*bad" + 0.013*well" +
'0.013*world" + 0.012*year" + 0.012*government" + 0.011*think" + '
'0.011*war'),
(3,
'0.017*forget" + 0.017*much" + 0.014*stop" + 0.013*israel" + 0.012*try" +
'+ 0.012*give" + 0.010*do" + 0.010*live" + 0.010*white" + 0.009*job'),
(4,
'0.022*kill" + 0.012*let" + 0.012*s" + 0.012*racist" + 0.012*state" +
'0.010*know" + 0.010*happen" + 0.009*help" + 0.008*culture" + '
'0.008*support')]
```



Key words distilled from topic identification could possibly serve as features to identify sarcasm. However, as discussed, this would need to be localised to individual topic groups and the effort for key word segregation could prove to be prohibitive in terms of computing time and cost.

## Stage 3 - Text-preprocessing

Text-preprocessing was carried out to convert text to lower case and remove unwanted white space.

Tokenization and Part of Speech Tagging (POS) was carried out using the Spacy NLP package.

Stop-word removal and lemmatization are considered further below.

### Stop-word removal:

As stated previously, the most-common word results indicate that filtering of stop words should be performed carefully, if at all, for the purposes of sarcasm identification. Stop words can contribute to context and their removal could result in a critical loss of information in NLP. For example, consider the 2 sentences below with “to” as the stop-word under consideration:

*“The boy gave the puppy to his mum.”*

*“The boy gave the puppy his mum.”*

Notwithstanding, the removal of stop words is considered essential for topic identification such as LDA, where their retention could result in the blurring of topics due to the unnecessary “noise” introduced without much topical contribution.

For this project, stop-words will be retained for general text-processing and filtered out prior to topic allocation studies.

### ***Lemmatization:***

Lemmatization was performed on both the “comment” and “parent\_comment” text to benefit Term Frequency Inverse Document Frequency (TFIDF) and Topic Identification analysis. The accuracy of these applications is dependent on a string match between words and this is achieved by lemmatizing their forms so that “all variants are consistent across documents. Lemmatization is also important for training word vectors, since accurate counts within the window of a word would be disrupted by an irrelevant inflection like a simple plural or present tense inflection.” [9]

## **Stage 4: Feature extraction and engineering**

The following features were extracted for conventional machine learning modelling including:

- word count
- character count
- punctuation count
- word density
- POS tagging for adverbs and adjectives
- Count vectorization (Bag of Words)
- TFIDF vectorization (Word and Character level, n-grams)

Additionally, the following engineered features were introduced:

### **1. Combined Count and TFIDF vectorization**

This was performed using the SKlearn feature union tool to create a concatenation of the vector features created using Count and TFIDF vectorization individually.

### **2. Fast-Text word embeddings**

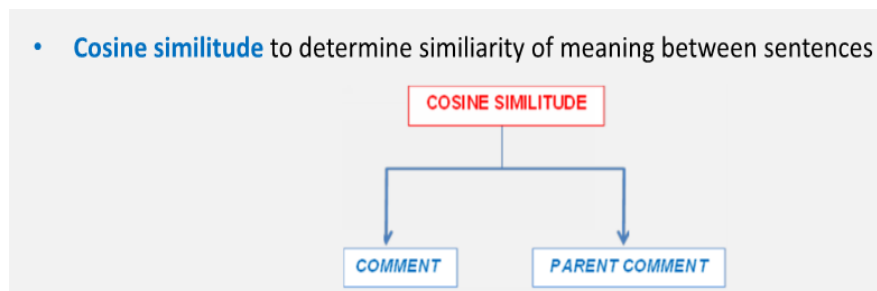
Word2Vec and Glove words embeddings do not account for words that do not exist in the model dictionary [10]. As much of speech on social media and online forums is informal, misspellings and abbreviations are commonplace, resulting in numerous words that do not exist in conventional dictionaries.

Fast-Text word embeddings were introduced by Facebook to address the above deficiency of Word2Vec and Glove word representations by accounting for the morphology (syntax and structure) of unknown or modified informal words. Words can even be broken down to their constituent n-grams to enable embeddings. [10] Hence, Fast-Text is considered most suitable for addressing Reddit forum postings, which are similar in characteristics to informal Facebook postings.

### 3. Cosine similitude comparison

For conventional machine learning approaches, the contextual information of the “parent\_comment” cannot be extracted easily. Simple concatenation of the “parent\_comment” with the “comment” risks introducing significant “noise” with little gain in predictive capability.

The measurement of cosine similitude between “comment” and “parent\_comment” is proposed in this project to indirectly account for the “parent comment”.

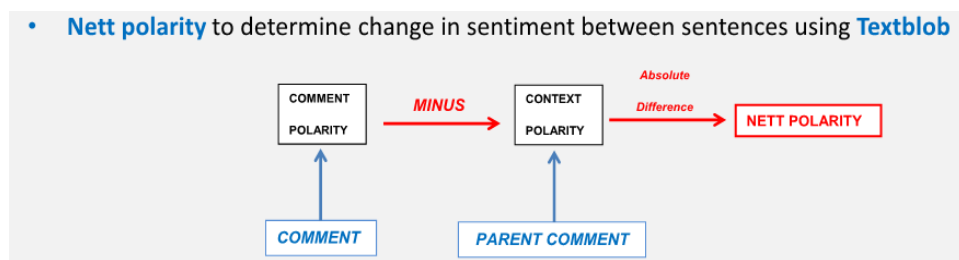


Cosine similitude is a potential marker for sarcasm as this measurement can detect changes in meaning between sentences, as would occur in contradiction arising from sarcastic expression.

Notwithstanding, this measurement will perform well only if there is a consistent pattern of dissimilitude occurring between “comment” and “parent comment”. For our dataset, varied expressions of sarcasm mean that sentence dissimilitude occur only in some, but not all sarcastic comments. As such, this inconsistency is expected to dilute the usefulness of cosine similitude as a predictive feature.

### 4. Polarity comparison

Comparison of polarity between “comment” and “parent\_comment” (using the TextBlob library) could provide another useful feature to train machine learning algorithms to detect polarity switching between sentences, which can occur for sarcastic expression.



Similar to observations for Cosine Similitude measurements, the inconsistency across sarcastic comments in the Reddit corpus (where some sentences demonstrate polarity flipping but others do not) would weaken the usefulness of this tool as a predictive feature.

The correlation of all extracted meta-features and engineered features (Excluding Count, TFIDF and FastText vectorization) with the target variable “label” may be examined using a heatmap plot of correlation coefficients, per below.

[illegible]

The figure consists of two side-by-side box plots. The left plot, titled 'Cosine similarity of each class', shows the distribution of cosine similarity for two classes: 0 (sarcastic, blue) and 1 (non-sarcastic, green). The y-axis ranges from -0.2 to 1.0. The right plot, titled 'Polarity of each class', shows the distribution of polarity for the same two classes. The y-axis ranges from 0.00 to 2.00. Both plots have a light gray background with white grid lines.

Class	Min	Q1	Median	Q3	Max
0 (sarcastic)	-0.25	0.08	0.18	0.30	1.00
1 (non-sarcastic)	-0.20	0.10	0.18	0.28	1.00

Class	Min	Q1	Median	Q3	Max
0 (sarcastic)	0.00	0.05	0.20	0.40	2.00
1 (non-sarcastic)	0.00	0.05	0.20	0.40	2.00

IOD Capstone Project by Yuan Chan 22/08/2021



## Stage 5A

3 x non-ensemble and 2 x ensemble (bagging and boosting) modelling techniques are employed at this stage.

The non-ensemble techniques (Logistic Regression, Naïve Bayes and Linear SVM) are selected for:

- Processing speed
- Simplicity of approach
- Interpretability of results

The ensemble techniques are higher performance algorithms that, in general, optimise prediction robustness over individual classifiers [11] per below:

- A minimum benefit of using ensembles is to reduce the spread in the average skill of a predictive model.
- A key benefit of using ensembles is to improve the average prediction performance over any contributing member in the ensemble.
- The mechanism for improved performance with ensembles is often the reduction in the variance component of prediction errors made by the contributing models

However, it should be noted that Bagging (Random Forest) decreases variance, not bias, and solves over-fitting issues in a model. On the other hand Boosting (Gradient Boosting) decreases bias, but not variance [12].

The introduction of both Random Forest and Gradient Boosting is anticipated to account for the above competing considerations.

## Stage 5B

The main focus of Stage 5B is to maximise the accuracy score of sarcasm predictions by combining the transformer architecture of a BERT encoder with the 2-way memory capacity of Bi-directional LSTM layers.

### BERT Encoder

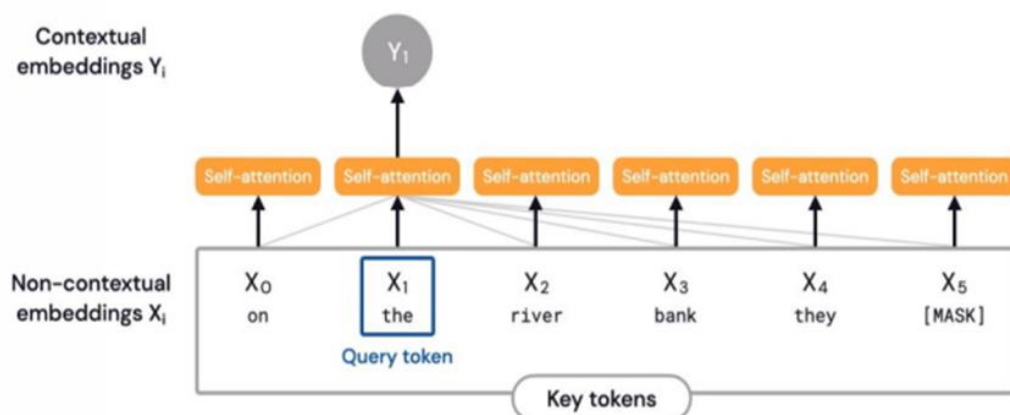
Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google [13].

The BERT BASE uncased model is employed in this project.

This model employs 12 Encoders with 12 bidirectional 'self-attention' heads [13] and does not differentiate between upper and lower cased English (hence 'uncased').

In BERT, 'self-attention' is the attribute that enables each token from the input sequence (e.g. sentences made of word or subword tokens) to focus on any other token.

## Self-Attention

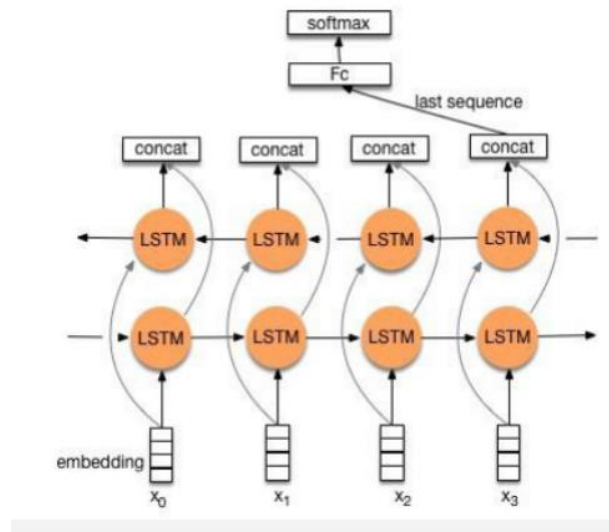


BERT 'self-attention' (Source: *Transfer Learning and Transformer Models*, ML Tech Talks, Tensorflow, Youtube)

## Bidirectional LSTM

Bidirectional LSTMs are extensions of traditional LSTMs that address text classification problems where sequencing is important.

Where all time-steps of the input sequence are available, “Bidirectional LSTMs train two instead of one LSTM on the input sequence. The first LSTM is positioned on the input sequence as-is and the second on a reversed copy of the input sequence. This twin approach accounts for both forward and backwards memory and can provide additional context to the network, resulting in faster and even fuller learning on the problem [14].



**Bi-directional LSTM architecture (Source: *Understanding Bidirectional RNN in PyTorch*, Ceshine Lee, Medium)**

Stage 5B is executed in 3 separate sub-stages:

- Sub-stage i: Basic feed forward multi-layer perceptron model (refer model architecture below - no pre-training, basic Keras word embeddings) – this forms the benchmark performance for comparison with the more complex BERT + BLSTM neural network.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 16)	160000
global_average_pooling1d (GlobalAveragePooling1D)	(None, 16)	0
dense (Dense)	(None, 24)	408
dense_1 (Dense)	(None, 1)	25
Total params: 160,433		
Trainable params: 160,433		
Non-trainable params: 0		

**Basic Feed-Forward Neural Network architecture**

- b) Sub-stage ii: BERT + BLSTM (BERT pre-training only) – Model adapted from Keras example for Natural Language Inference (NLI). The Keras model is trained for multi-class identification of the three semantic categories “contradiction”, “entailment” and “neutral”. The model is adapted to suit binary classification and trained entirely on the Reddit dataset. In this instance, the only transfer learning is via the pre-trained BERT transformer encodings.

The BERT + BLSTM model architecture consists of the following 10 layers with final ‘Softmax’ activation for the output layer.

For compilation of the model, an ‘Adam’ optimizer with ‘Binary\_crossentropy’ class loss function is adopted.

**Model**

**Neural network model architecture**

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 128)]	0	
attention_masks (InputLayer)	[(None, 128)]	0	
token_type_ids (InputLayer)	[(None, 128)]	0	
tf_bert_model (TFBertModel)	((None, 128, 768), ( 109482240		input_ids[0][0] attention_masks[0][0] token_type_ids[0][0]
bidirectional (Bidirectional)	(None, 128, 128)	426496	tf_bert_model[0][0]
global_average_pooling1d (GlobalAveragePooling1D)	(None, 128)	0	bidirectional[0][0]
global_max_pooling1d (GlobalMaxPooling1D)	(None, 128)	0	bidirectional[0][0]
concatenate (Concatenate)	(None, 256)	0	global_average_pooling1d global_max_pooling1d
dropout_37 (Dropout)	(None, 256)	0	concatenate[0][0]
dense (Dense)	(None, 2)	514	dropout_37[0][0]

Total params: 109,909,250  
Trainable params: 427,010  
Non-trainable params: 109,482,240

**BERT + BLSTM Neural Network architecture**

- c) Sub-stage iii: BERT + BLSTM (NLI classification pre-training + BERT fine-tuning) – For this stage, the model is pre-trained for NLI classification but the model architecture is subsequently modified for binary classification. After convergence on the NLI data is achieved, the BERT model is “unfrozen” to render it “trainable”. The model is then re-trained entirely on the Reddit dataset at a very low learning rate (1e-5), enabling the BERT encoder to fine-tune word embeddings based on the new data.

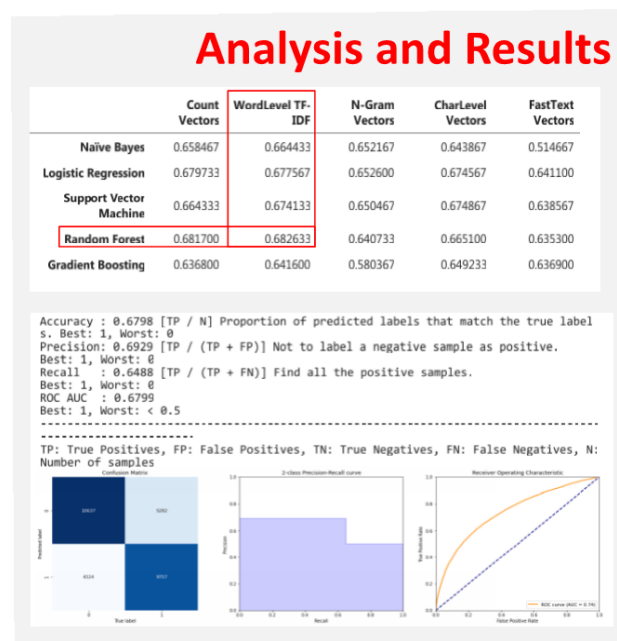
## Stage 6 - Model Evaluation and discussion of results

### Stage 5A results

Results obtained using conventional machine learning and text classification approaches (WordLevel TF-IDF and Random Forest classifier) achieves the following scores:

- Test accuracy score of **0.68**
- Precision score of 0.69
- Recall score of 0.65
- Area Under curve (AUC) ratio of 0.74 for the Receiver Operator Characteristic (ROC) curve.

There is no evidence of significant overfitting with a training accuracy score of 0.655 following hyper-parameter tuning.



The achieved accuracy score is significantly above the baseline accuracy of 0.50.

## Stage 5B results

Refer to below table for Stage 5B results:

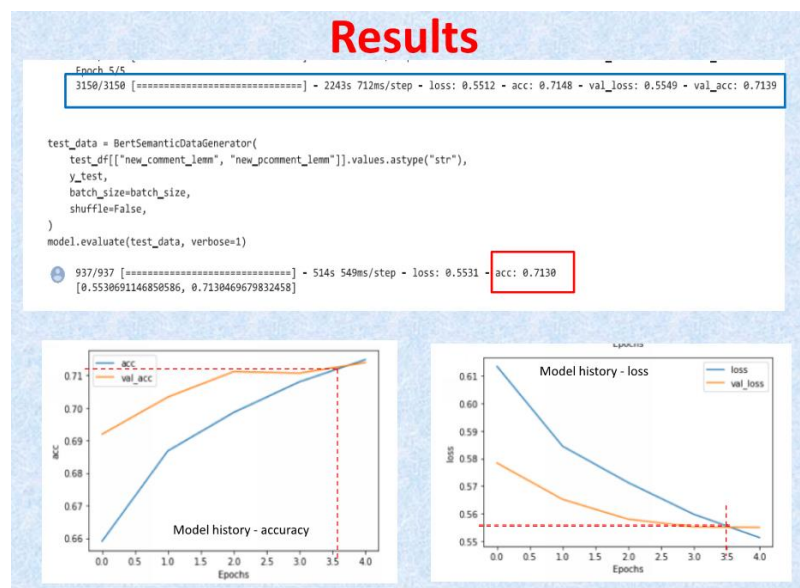
### Substage i – Feed forward Multi-level perceptron

A benchmark accuracy score of **0.67** is attained on the training validation data.

### Substage ii – BERT + BLSTM without NLI fine-tuning

Without NLI fine tuning, an accuracy score of **0.713** is achieved on the test data (5 epochs).

Learning curves indicate that circa 3 epochs produce optimised model results in terms of test accuracy and minimisation of overfitting.

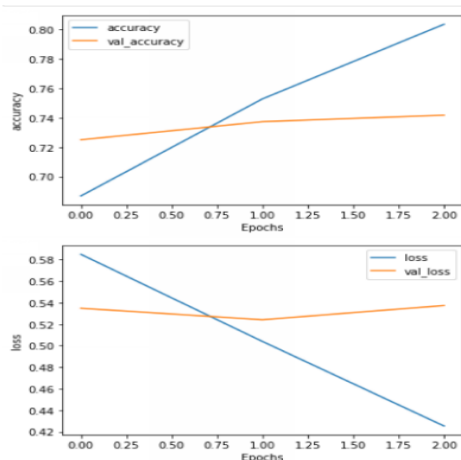


### Substage iii – BERT + BLSTM with NLI fine-tuning

With pre-training on an external Natural Language Inference (NLI) dataset for classification of “contradiction” vs “Entailment” and fine-tuning the BERT model using a very small learning rate (1e-5) on the Reddit dataset, an accuracy score on the test data of **0.74** is achieved.

This represents a significant improvement of 10% over the score attained by the Stage 5A Random Forest classifier combined with conventional text classification approaches.

```
model.evaluate(new_test_data, verbose=1)
937/937 [=====] - 492s 525ms/step - loss: 0.5415 - accuracy: 0.7401
: [0.5415017008781433, 0.7400947213172913]
```



### Test predictions

For illustrative purposes, test predictions on sarcasm detection using the BERT + BLSTM model are performed on a set of sample sentences (sentence 1 is the ‘comment’ and sentence 2 the ‘parent comment’), per below:

```
sentence1 = "i cant access my centrelink account"
sentence2 = "are you going to be ok"
check_sarcasm(sentence1, sentence2)

('Not Sarcastic', ' 0.80%')

sentence1 = "covid is not real"
sentence2 = "thousands of people are dying without access to healthcare"
check_sarcasm(sentence1, sentence2)

('Sarcastic', ' 0.56%')

sentence1 = "hes so skinny"
sentence2 = "no he's not...he is the fattest person in a third world country"
check_sarcasm(sentence1, sentence2)

('Sarcastic', ' 0.64%')
```

The results demonstrate the ability of the model to correctly identify sarcasm on all three examples.

## CONCLUSIONS AND FUTURE DIRECTION

This project has demonstrated 2 different tiers of machine learning modelling approaches for the NLP task of detecting sarcasm in text:

- a. Conventional machine learning approaches incorporating traditional text classification
- b. Deep learning neural networks incorporating transformer architecture and bi-directional RNN layers (BERT + BLSTM)

Deep-learning (BERT + BLSTM) accuracy scores were demonstrated to be at least 10% more accurate than conventional machine learning approaches, with accuracy scores lifted from 68% to 74%.

This improvement is attributable to the superior architecture inherent in the BERT + BLSTM models for accounting for sentence sequencing and extraction of sentence context.

It is notable that the score of 74% was attained on a dataset that had significant “noise” due to self-labelling of the target variable. Higher predictive scores are anticipated on ‘cleaner’ datasets that have been independently labelled (such as by a trained linguist).

A larger subset of the original database is also anticipated to boost accuracy scores as the Deep Learning model will generalise better.

Future investigations would entail extending pre-training on the Deep Learning models to other external datasets, such as those pre-trained on the following similar tasks:

- Sentiment analysis
- Personality analysis
- Fake news classification

If successful, such varied pre-training may improve the generalisation of the final classification model to predict across a broad range of datasets for classification of sarcastic content.



## REFERENCES

1. Poria, S., Cambria, E., Hazarika, D., Vij, P., 2017, *A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks*, Nanyang Technological University, Singapore, arXiv:1610.08815v2 [cs.CL],
2. Hallac, I.R., Ay, B., Aydin, G., 2021, User Representation Learning for Social Networks: An Empirical Study, *Applied Sciences (Journal)*, Department of Computer Engineering, Firat University, 23119 Elazig, Turkey
3. Merriam Webster Dictionary, *Sarcasm*, accessed August 2021, <https://www.merriam-webster.com/dictionary/sarcasm>
4. Reddit help forum, accessed August 2021, [https://www.reddit.com/r/help/comments/acls8e/what\\_does\\_s\\_mean\\_in\\_reddit/](https://www.reddit.com/r/help/comments/acls8e/what_does_s_mean_in_reddit/)
5. Ofer, D. *Sarcasm on Reddit*, accessed August 2021, Kaggle Website, <https://www.kaggle.com/danofer/sarcasm>
6. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D., 2015, *A large annotated corpus for learning Natural Language Inference*, Stanford University, arXiv:1508.05326 [cs.CL]
7. Merchant, M., 2015, *Semantic Similarity with BERT*, Keras, accessed August 2021, [https://keras.io/examples/nlp/semantic\\_similarity\\_with\\_bert/](https://keras.io/examples/nlp/semantic_similarity_with_bert/)
8. Matthews, G., Gilliland, K., 1999, *The personality theories of H.J. Eysenck and J.A. Gray: A comparative review*, *Personality and Individual differences* 26(4):583-626
9. Schumacher, A., 2019, *When (not) to Lemmatize or Remove Stop Words in Text Preprocessing*, accessed August 2021, <https://opendatagroup.github.io/data%20science/2019/03/21/preprocessing-text.html>
10. Kathrani, K. 2020, *All about Embeddings*, accessed August 2021, <https://medium.com/@kashyapkathrani/all-about-embeddings-829c8ff0bf5b>
11. Brownlee, J., 2021, *Why use ensemble learning?*, Machine Learning Mastery, accessed August 2021, <https://machinelearningmastery.com/why-use-ensemble-learning/>
12. Vadapalli, P., 2020, *Bagging Vs Boosting in Machine Learning*, Upgrad, accessed August 2021, <https://www.upgrad.com/blog/bagging-vs-boosting/>
13. Wikipedia entry, *BERT (Language Model)*, accessed August 2021, [https://en.wikipedia.org/wiki/BERT\\_\(language\\_model\)](https://en.wikipedia.org/wiki/BERT_(language_model))
14. Brownlee, J., 2021, *How to Develop a Bidirectional LSTM For Sequence Classification in Python with Keras*, Machine Learning Mastery, accessed August 2021, <https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/>