Sarcasm Detection

The Gateway to Sentiment Analysis

Team 10

Ramalingam Saravanamani
Liao Keng I
Pranav Venkatram
Wei Yangken
Shyam Ganesh Jayagopi
Ong Ting Rui, Brandon

01

Project Motivation

Sarcasm Detection in the context of Sentiment Analysis

What is Sarcasm?

The use of remarks that clearly mean the opposite of what they say, made in order to hurt someone's feelings or to criticize something in a humorous way

Cambridge Dictionary

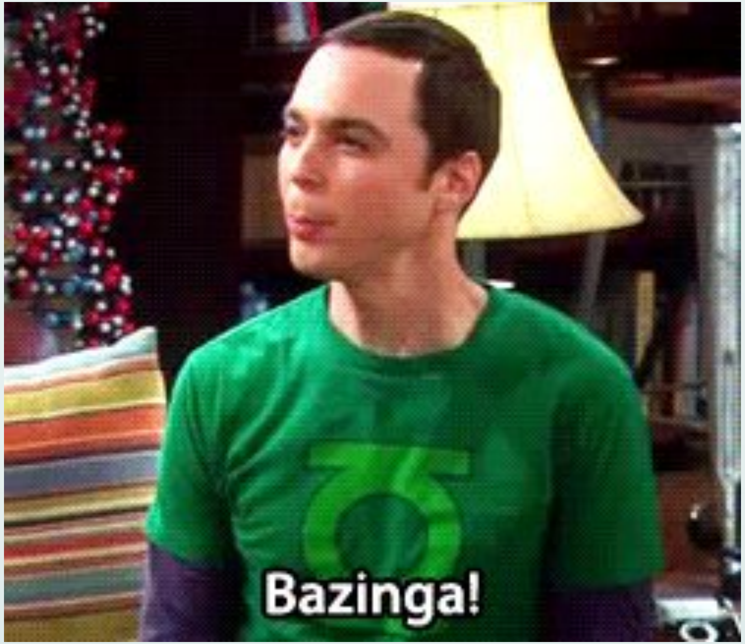
What is Sarcasm?

The use of remarks that clearly mean the opposite of what they say, made in order to hurt someone's feelings or to criticize something in a humorous way

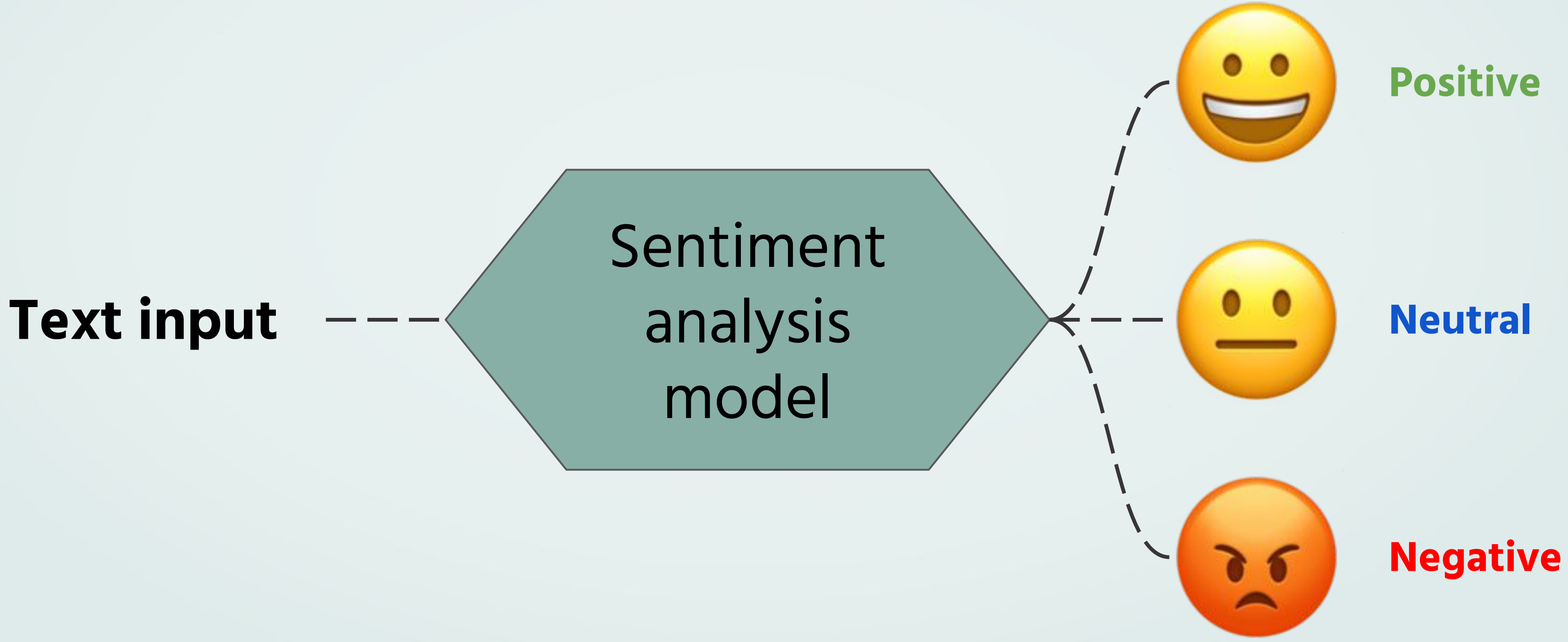
Cambridge Dictionary

What is Sarcasm?

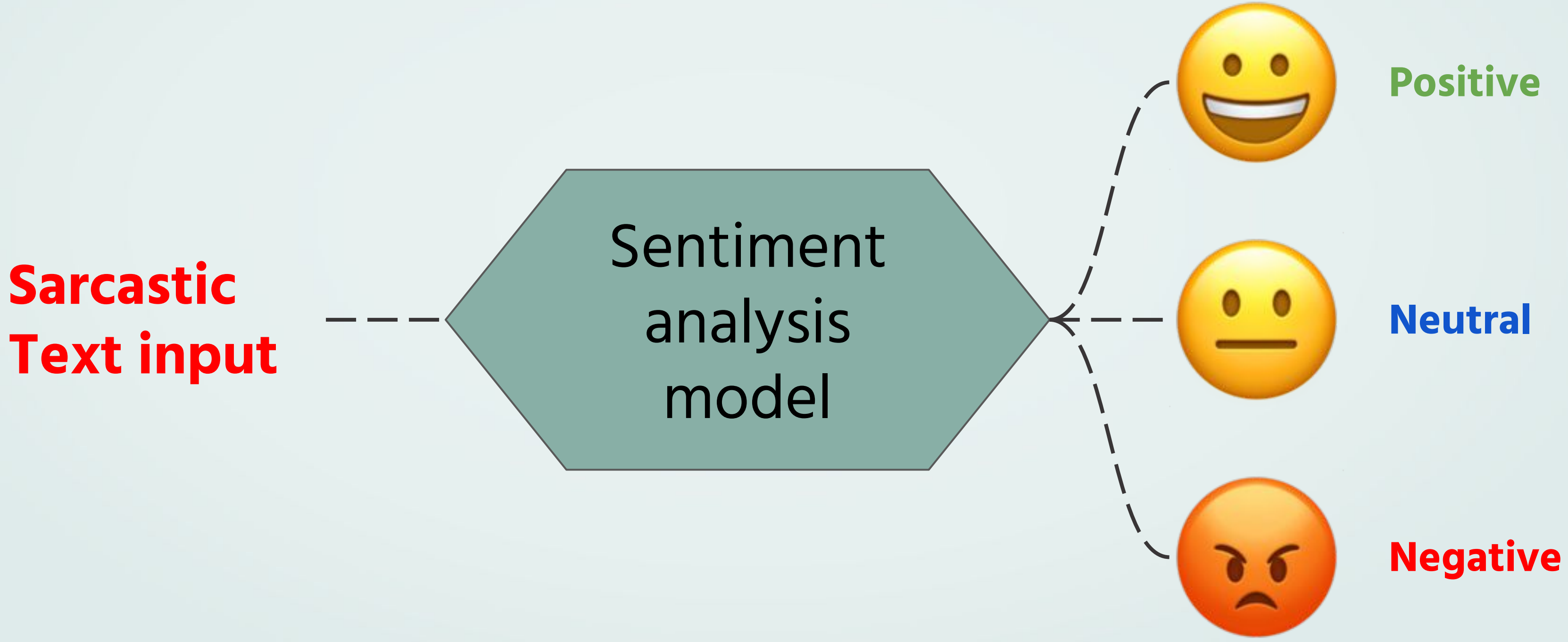
Mean the opposite of what is on the surface



Problem of Sarcasm in Sentiment Analysis



Problem of Sarcasm in Sentiment Analysis



Problem of Sarcasm in Sentiment Analysis

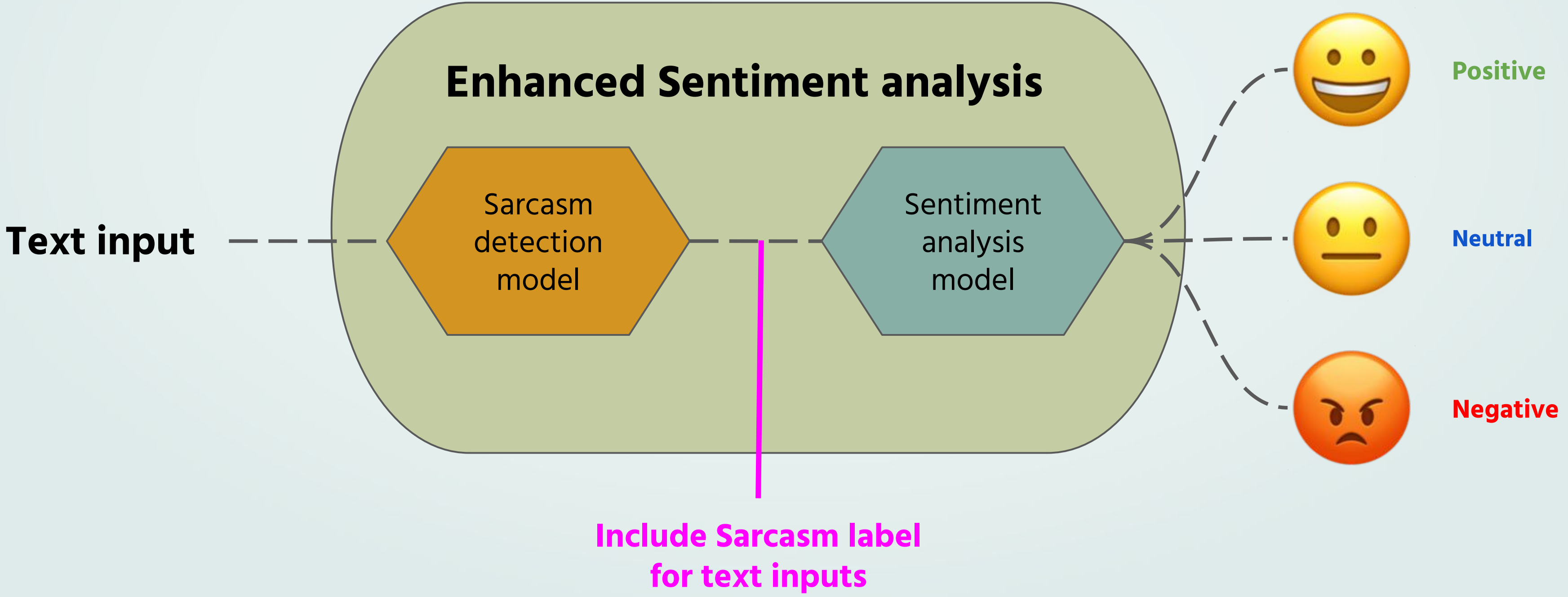
Given the sentence..

Wow! Sarcasm detection using ML is **sooooooo easy**, even a baby can do it!!!!!

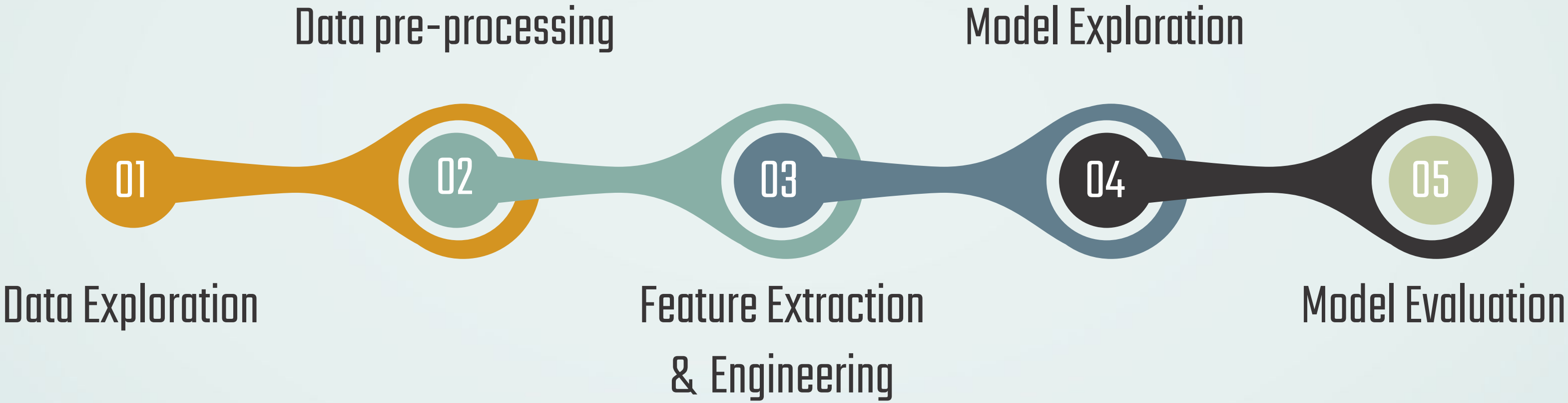
Unaware of Sarcasm -> Positive

Enhanced Sentiment analysis

Enhanced Sentiment analysis



General Approach



Data Exploration

02

Dataset Columns

Comment
Given text to check for sarcasm

Parent Comment
Text that comment is replying to

Score
Number of upvotes minus the number of downvotes

Author
Writer of the comment

SubReddit
General topic that comment falls under

Created Time
Time that comment was posted

Dataset Columns

Comment
Given text to check for sarcasm

~~**Parent Comment**~~
Text that comment is replying to

~~**Score**~~
Number of upvotes minus the number of downvotes

~~**Author**~~
Writer of the comment

~~**SubReddit**~~
General topic that comment falls under

~~**Created Time**~~
Time that comment was posted

Columns dropped as the text to be predicted on **does not include these features**

Exploration of ***Comment*** column

Missing Data

53 rows of data with comments **missing**

Duplicated Comments

48479
repeated comments

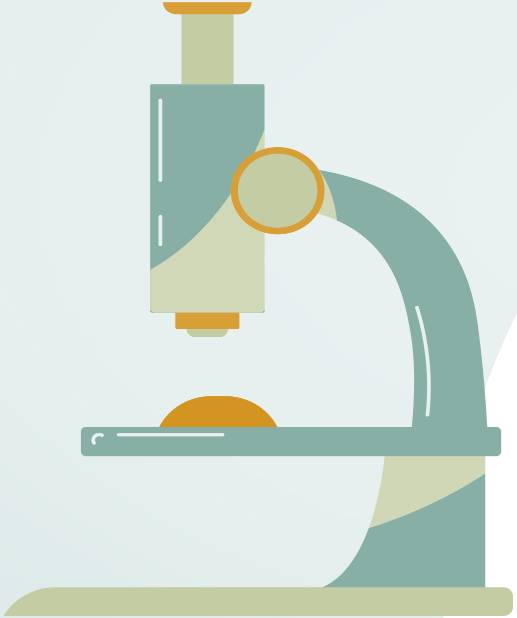
Most common words

the, a, to, it, i, and, you, is, of, that, in

Numerical Comments

1277 numerical comments after removing punctuation

Common text-preprocessing techniques



- Drop rows with null values
- Remove punctuations
- Lowercase all text
- Tokenize into words
- Remove stopwords
- Stem words
- Lemmatize words
- Drop numerical data

Pre-processing effects

Original Comment

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

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

1 Remove punctuation

2 Lowercase all text

how about a no lives matter for the incurably misanthropic

[how, about, a, no, lives, matter, for, the, incurably, misanthropic]

3 Tokenize into words

4 Remove safe stopwords

[how, ~~about~~, ~~a~~, no, lives, matter, ~~for~~, ~~the~~, incurably, misanthropic]

[how, no, live, matter, incurably, misanthropic]

5 Stem words

6 Lemmatize words

[how, no, life, matter, incurably, misanthropic]

[how, no, life/live, matter, incurably, misanthropic]

7 Drop if comment is numeric

Cleaning data may remove features! 🙁

Pre-processing impact on sarcasm detection

Removing punctuation

Comments may use punctuation to express the idea of so-called [1], which implies sarcasm
Eg. How are you so "smart"......?

Removing stopwords

Context will shift if stopwords that contribute to sentiment are missing. To difficult to predict all possibilities
Eg. He is totally not feeling sick

[1] https://style.mla.org/scare-quotes-origins/

Pre-processing impact on sarcasm detection

Stemming

Stemming is a technique used to contract words to their roots forms. For instance, connections, connected, connects, stems to “connect”.

Lemmatization

Lemmatization is another text normalization technique to change words to their base root form. For instance, the words leafs and leaves become “leaf”.

Why is it a problem?

The extent of a word and its implication is lost. The “smart~~est~~” man is no longer the “smartest”!

Feature Extraction & Engineering

03

Vectorizers and Word embedding

Bag of Bigrams

TF.IDF

GloVe word embedding

Keras tokenizer

Features from Literature [2]

Frequencies of:

1. Consecutive Alphabets
2. Exclamation Marks
3. Dots
4. Question Marks
5. Capital Letters
6. Quotation Marks

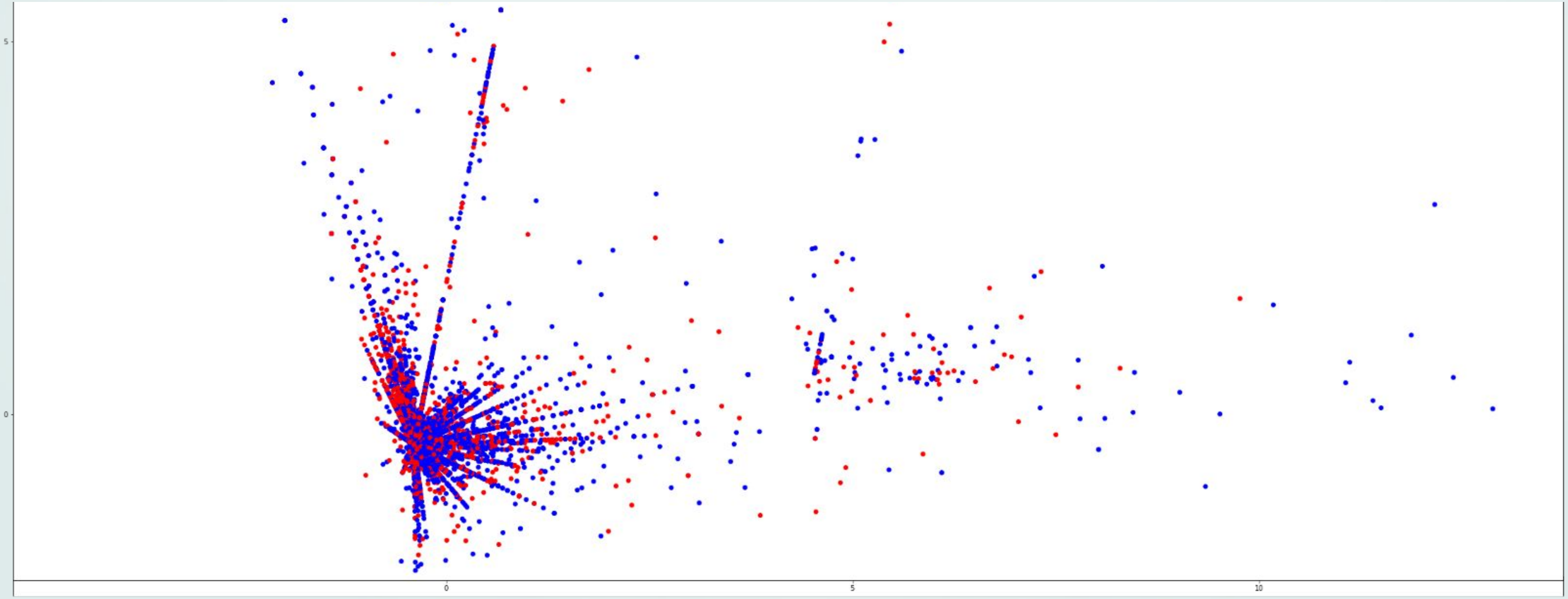
[2]https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3384025

Correlation Matrix

PCA plot of Literature Features

- Non-sarcastic

- Sarcastic



Cropped plot for viewing important region

New Augmented Features

Frequencies of:

1. Consecutive Alphabets (same as literature)
2. Consecutive Exclamation Marks
3. Consecutive Dots
4. Consecutive Question Marks
5. Consecutive Capital Letters
6. Consecutive Punctuations

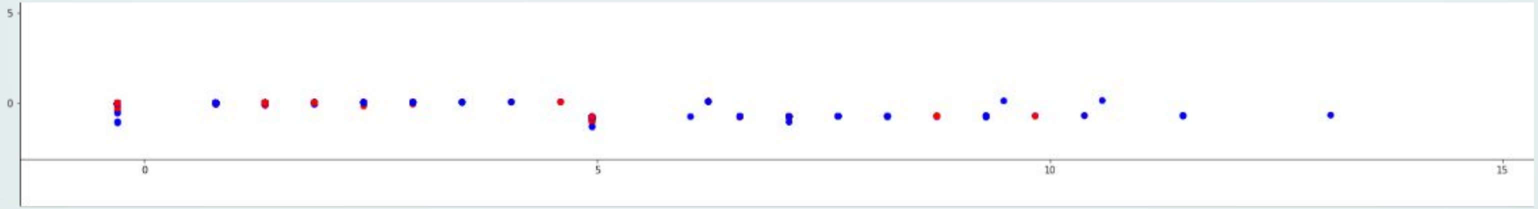
Consecutive: More than 2 occurrences in a row

Correlation Matrix of New Features

PCA plot of New Features

- Non-sarcastic

- Sarcastic

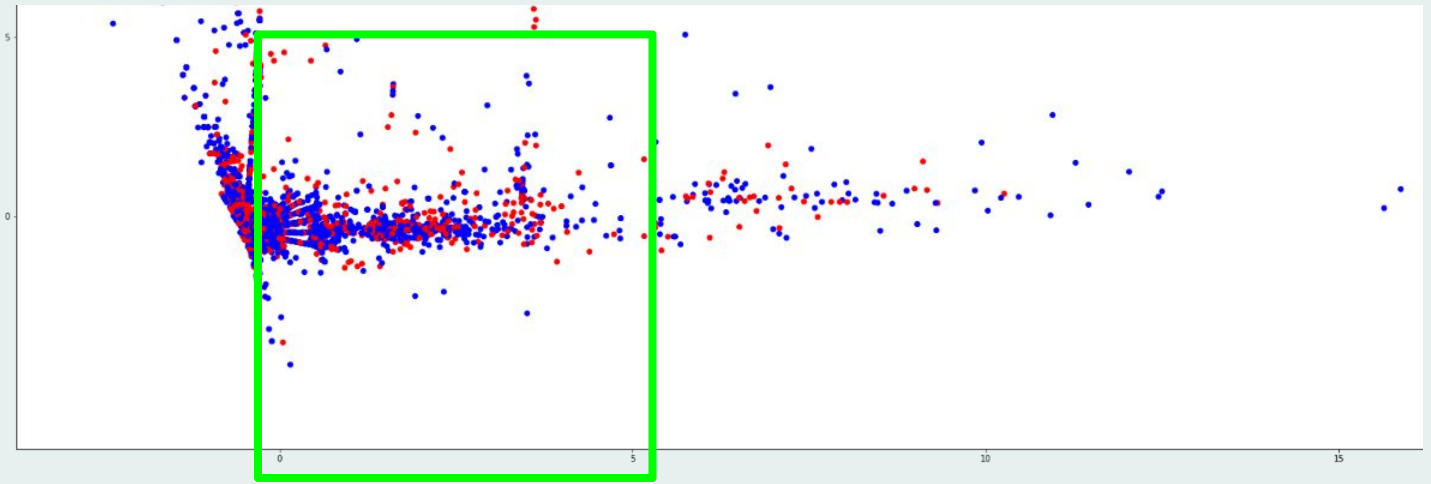


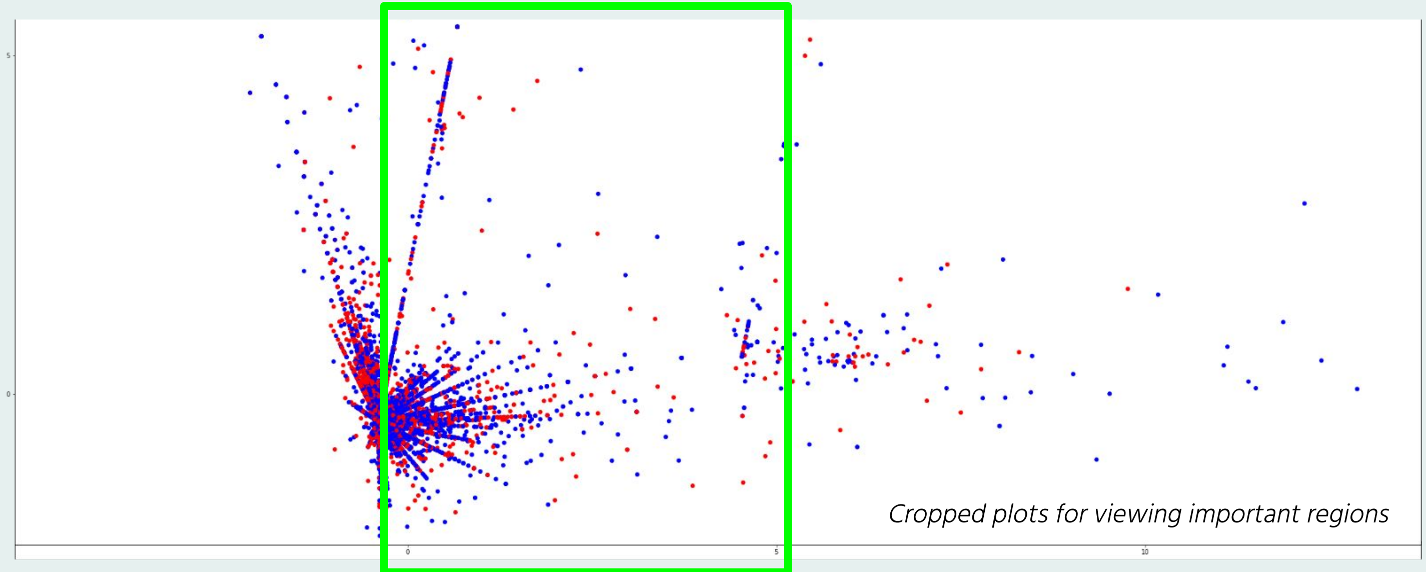
Cropped plot for viewing important region

PCA plot of Combined Features

VS

PCA plot of Literature Features





Cropped plots for viewing important regions

04

Model Exploration

Models Explored

1. Logistic Regression
2. Recurrent Neural Networks (RNN)
3. Convolutional Neural Network (CNN)
4. Bidirectional Encoder Representations from Transformers (BERT)

LOGISTIC REGRESSION

Motivation & Architecture

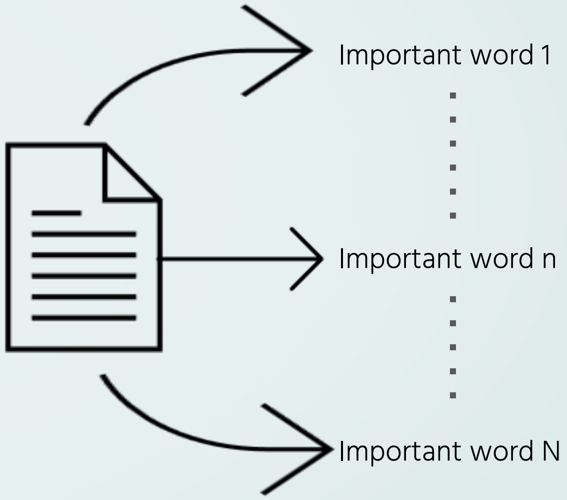
- **Large reddit dataset** thus using logistic regression would **reduce training time significantly**
- TF-IDF outputs words which are **most representative of sarcasm (using 200 most frequent words as vocabulary)** [2]

[2]https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3384025

LOGISTIC REGRESSION

Motivation & Architecture

- **More than 1 million data points** thus using logistic regression would reduce training time significantly
- TF-IDF outputs words which are **most representative of sarcasm (using 200 most frequent words as vocabulary) [2]**

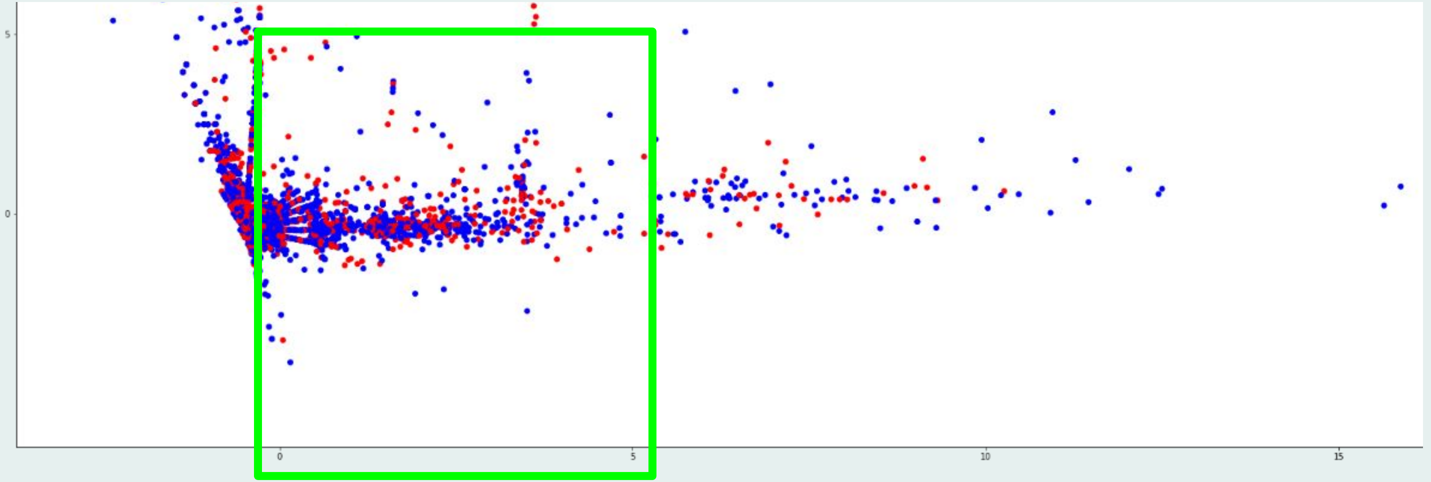


[2]https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3384025

Results

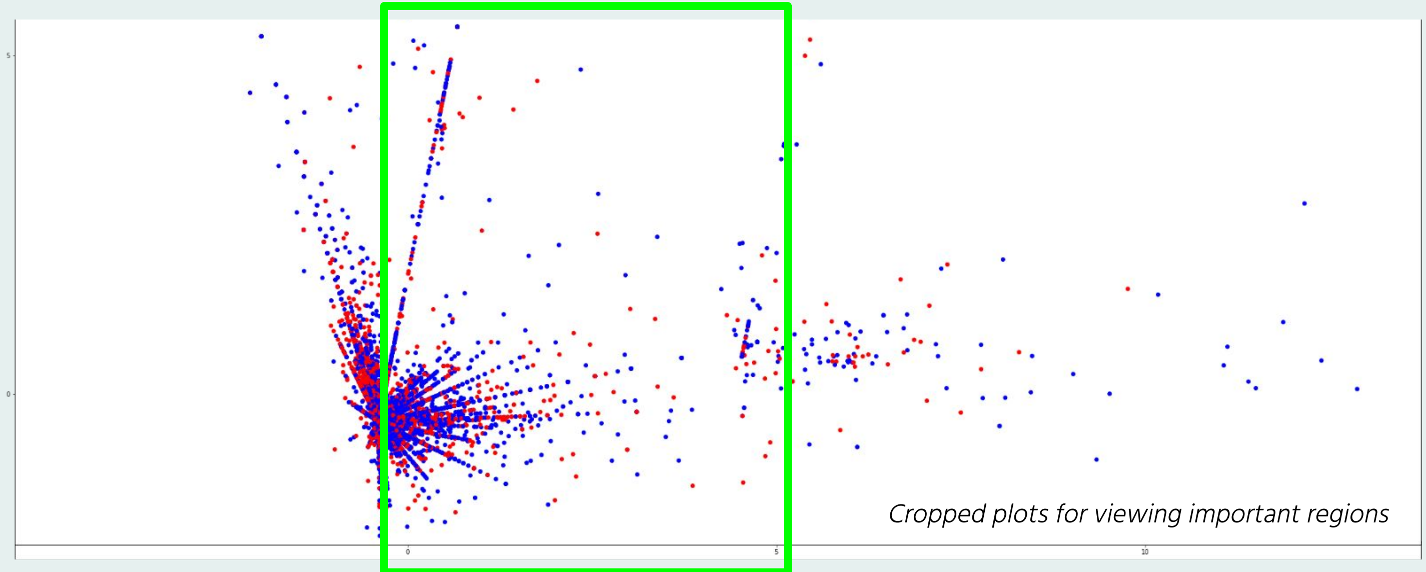
	Accuracy	Precision	Recall	F1 score
Features from paper	63.55 %	0.659	0.562	0.607
Combined with custom features	63.99%	0.654	0.595	0.623

PCA plot of Combined Features



VS

PCA plot of Literature Features



Cropped plots for viewing important regions

Models Explored

1. Logistic Regression
2. **Recurrent Neural Networks (RNN)**
3. Convolutional Neural Network (CNN)
4. Bidirectional Encoder Representations from Transformers (BERT)

Recurrent Neural Networks (RNN)

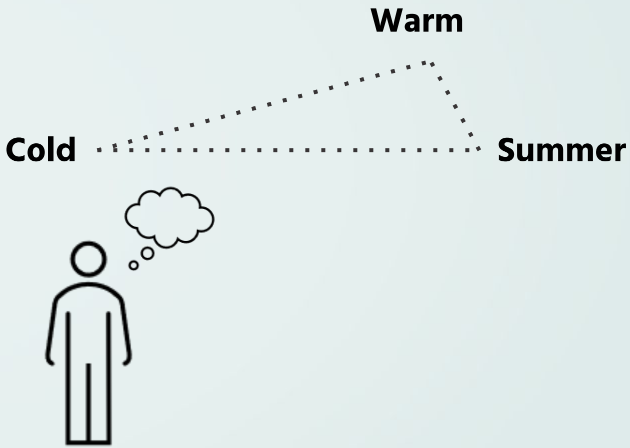
Motivation & Architecture

- **More than 1 million data points** encourages deep learning with Neural Networks
- **GloVe** allows us to understand **vector embeddings of words**
- **LSTM** allows us to keep track of **long term and short term memory**

Recurrent Neural Networks (RNN)

Motivation & Architecture

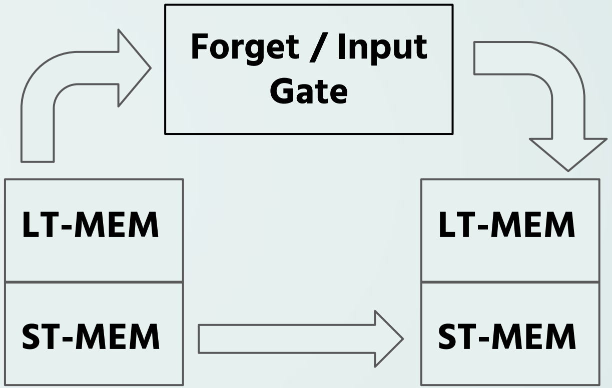
- **More than 1 million data points** encourages deep learning with Neural Networks
- **GloVe** allows us to understand **vector embeddings of words**
- **LSTM** allows us to keep track of **long term and short term memory**



Recurrent Neural Networks (RNN)

Motivation & Architecture

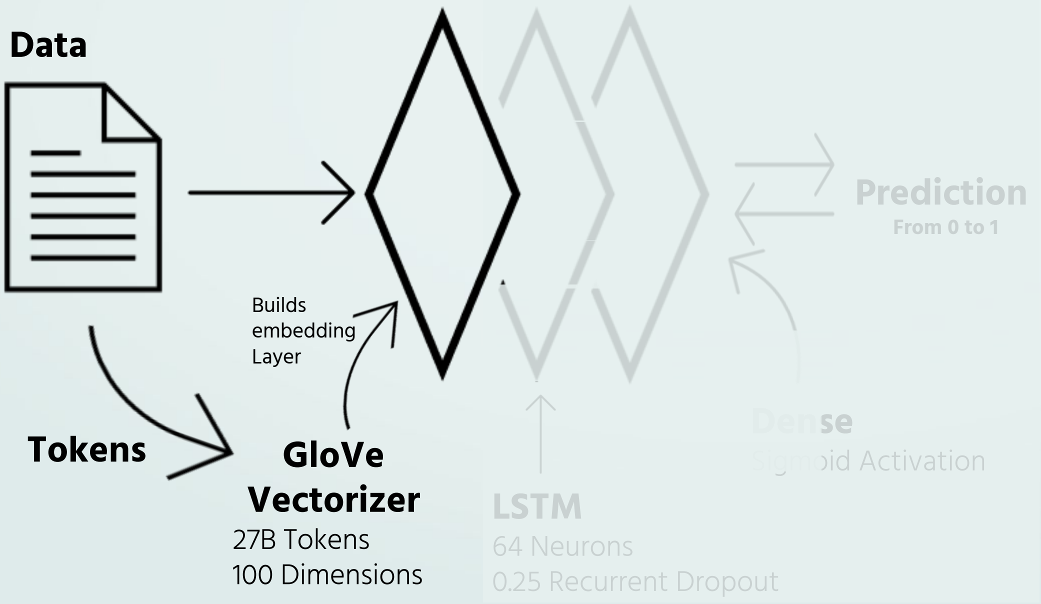
- **More than 1 million data points** encourages deep learning with Neural Networks
- **GloVe** allows us to understand **vector embeddings of words**
- **LSTM** allows us to keep track of **long term and short term memory**



Forget and Input Gates allows for **removal** and **insertion** for important keywords respectively in our **Long-Term Memory**.

Recurrent Neural Networks (RNN)

Baseline Model Structure

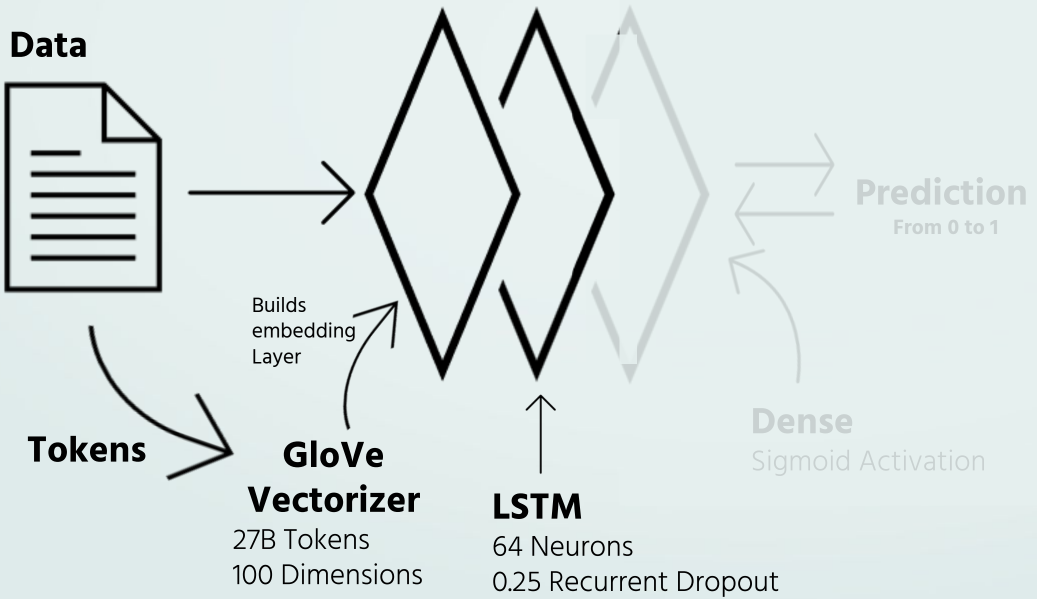


Rationale

GloVe vectorizes the word tokens to build an **embedding layer of 100 dimensions**.

Recurrent Neural Networks (RNN)

Baseline Model Structure

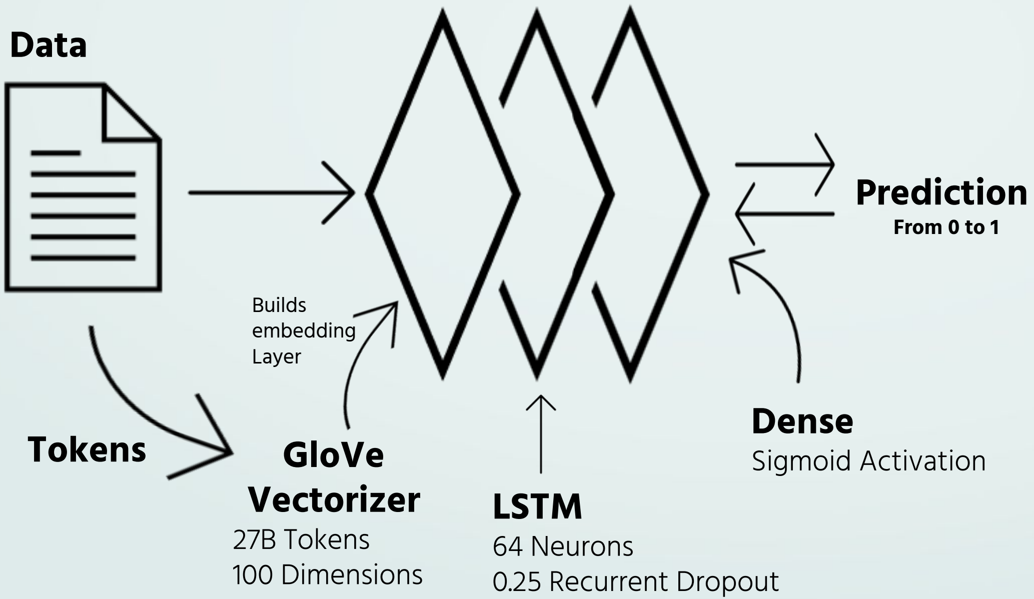


Rationale

LSTM stores the **important keywords** and captures **relationships** between words.

Recurrent Neural Networks (RNN)

Baseline Model Structure



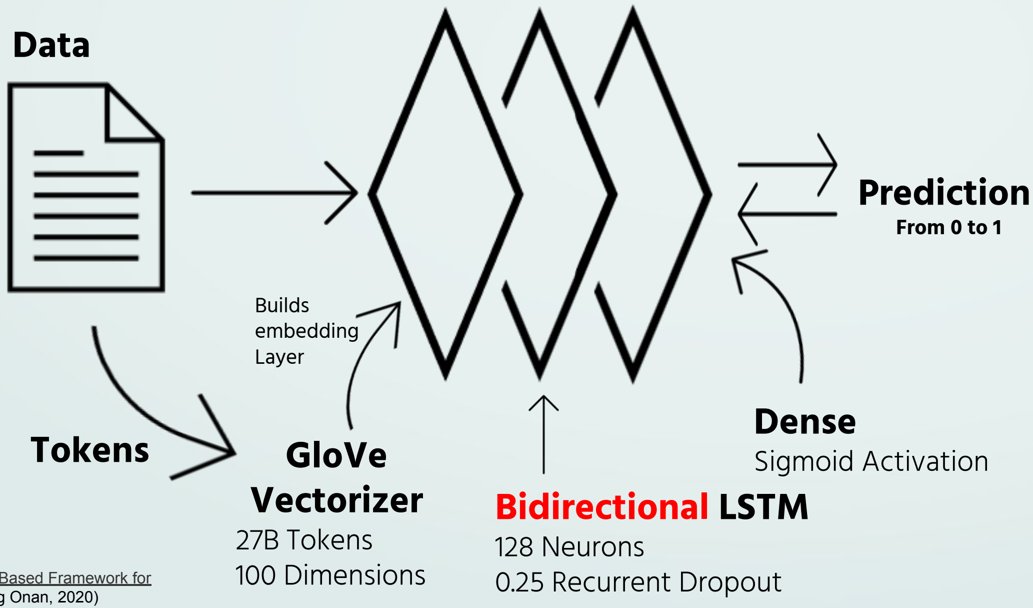
Rationale

Sigmoid function maps our output to a **sarcasm** probability **from 0 to 1**.

	Acc.	Prec.	Recall	F1
Baseline	70.6%	0.717	0.682	0.700

Recurrent Neural Networks (RNN)

Augmented Model Structure



Rationale

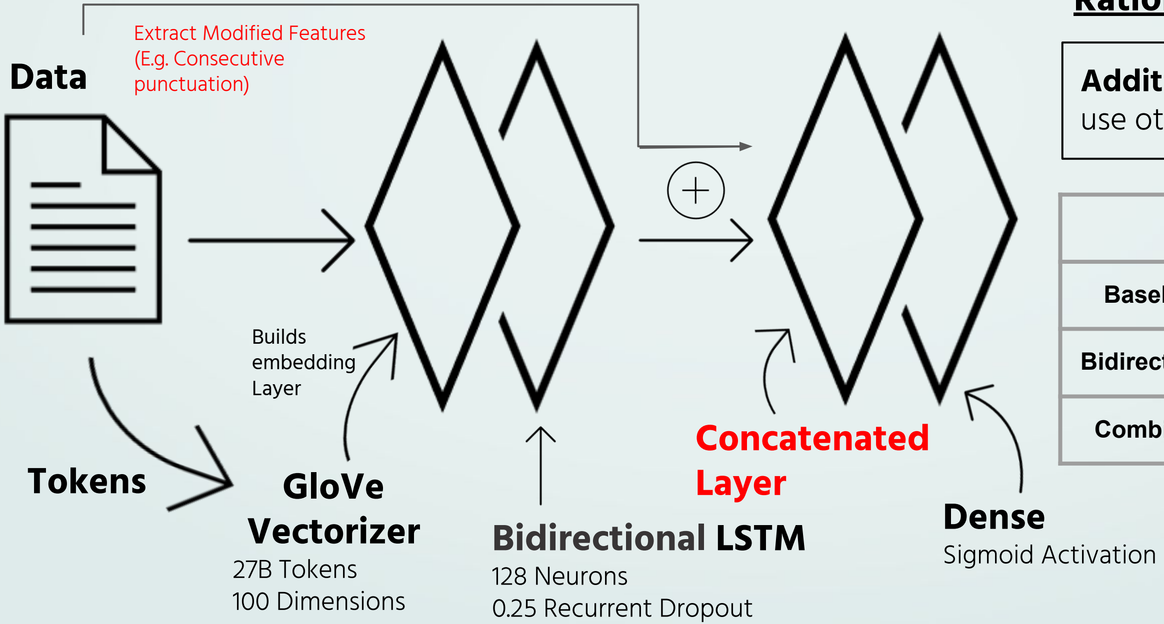
Bidirectional LSTM learns the **reversed** sequencing for **pattern recognition**.

	Acc.	Prec.	Recall	F1
Baseline	70.6%	0.717	0.682	0.700
Bidirectional	71.6%	0.731	0.683	0.707

Stacked Bidirectional LSTM Based Framework for Sarcasm Identification (Aytug Onan, 2020)

Recurrent Neural Networks (RNN)

Augmented Model Structure



Rationale

Addition of features allow our model to use other indicators of sarcasm.

	Acc.	Prec.	Recall	F1
Baseline	70.6%	0.717	0.682	0.700
Bidirectional	71.6%	0.731	0.683	0.707
Combined	71.8%	0.755	0.683	0.700

2400 more correctly classified instances.

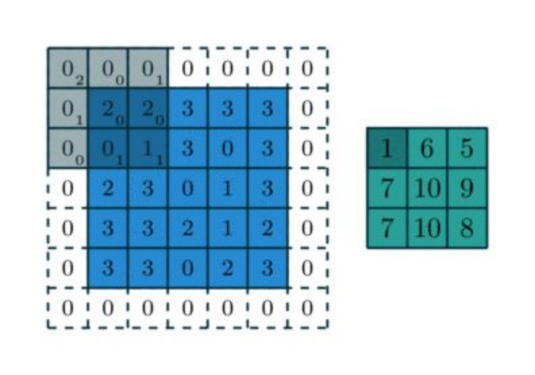
Models Explored

1. Logistic Regression
2. Recurrent Neural Networks (RNN)
3. **Convolutional Neural Network (CNN)**
4. Bidirectional Encoder Representations from Transformers (BERT)

Convolutional Neural Networks (CNN)

Motivation & Architecture

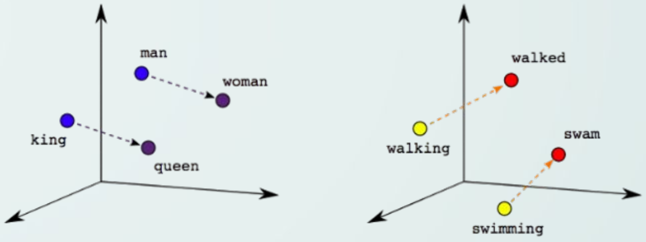
- **CNN** uses convolutions along with non-linear activation function like ReLU to capture sentiments within sentences
- **Word embedding** enables to represent text with similar meaning
- **Convolution Filters and Hidden layers** of neural network act as a feature extractor for the word vectors



Convolutional Neural Networks (CNN)

Motivation & Architecture

- **CNN** uses convolutions along with non-linear activation function like ReLU to capture sentiments within sentences
- **Word embedding** enables to represent text with similar meaning
- **Convolution Filters and Hidden layers** of neural network act as a feature extractor for the word vectors
-



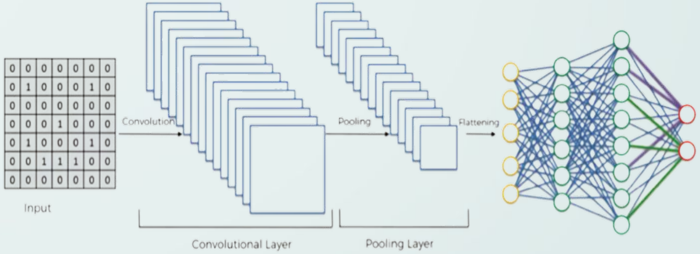
Male-Female

Verb Tense

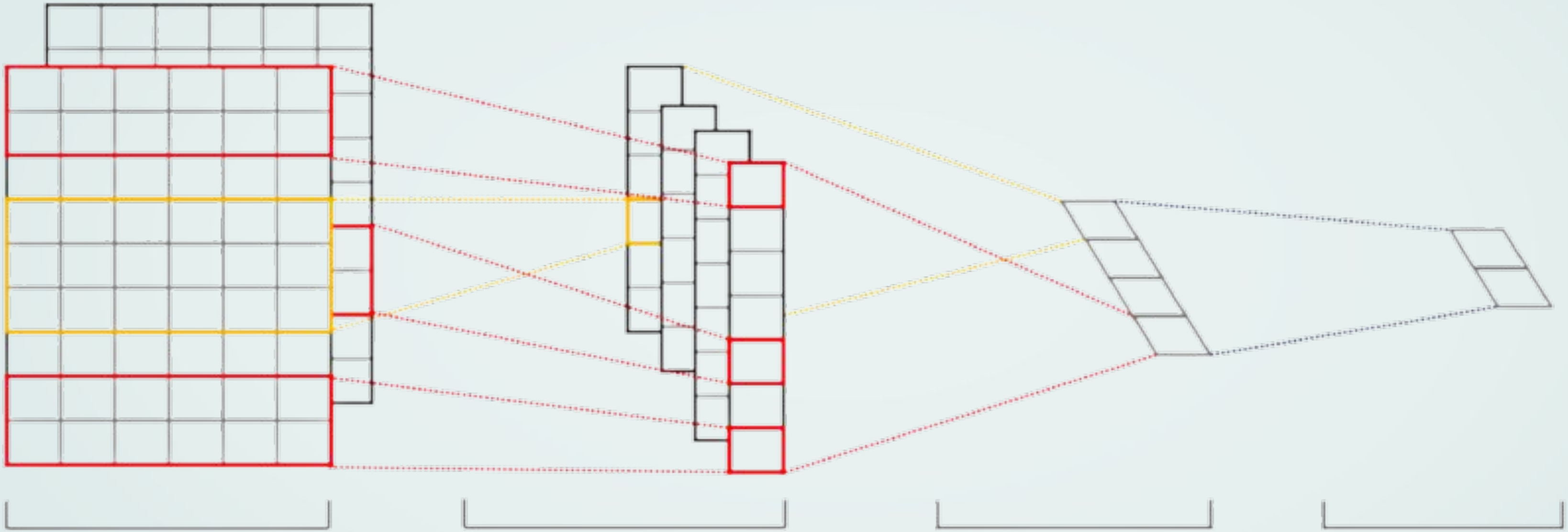
Convolutional Neural Networks (CNN)

Motivation & Architecture

- **CNN** uses convolutions along with non-linear activation function like ReLU to capture sentiments within sentences
- **Word embedding** enables to represent text with similar meaning
- **Convolution Filters and Hidden layers** of neural network act as a feature extractor for the word vectors



CNN Model Architecture



Embedding Layer
Vector space - 50000
Output vector - 64
Input length - 75

Conv1d Layer
Output filters - 256
Kernel size - 5

GlobalMaxPooling 1d Layer
Dropout layer

Dense Layers
Activation Function - ReLU, Sigmoid
Units - 512, 1

Tuning Hyperparameters

Choose the parameters to optimize

For our CNN model we decided to optimize the hidden layers, embedding layers, kernel size, filters, batch size, learning rate and dropout

Hyperparameter search

After selecting the hyperparameters we decided to use random search to get our hyperparameters. Random search although might not give the best set of hyperparameters it is still much faster than grid search

Tuning Hyperparameters

Optimal parameters obtained

Hidden Layers	Embedding Layers	Kernel Size	Filters	Batch Size	Learning Rate	Dropout
512	64	5	256	32	0.0006	0.05

Results

	Accuracy	Precision	Recall	F1 score
Before Tuning	73.02 %	0.786	0.680	0.700
After Tuning	73.68 %	0.761	0.689	0.723
With Custom Features	74.21 %	0.758	0.712	0.734

Results

Data pre processing type	Accuracy	Precision	Recall	F1 Score
Preprocessing data	0.719607	0.716061	0.731468	0.723682
Without preprocessing data	0.736848	0.761048	0.689230	0.723361
With Custom Features	0.742163	0.758208	0.712466	0.734626

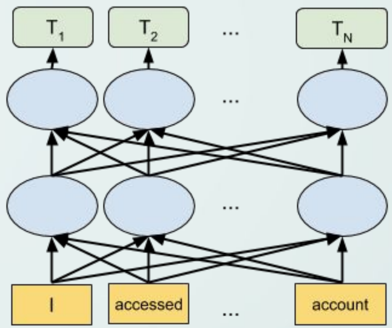
Models Explored

1. Logistic Regression
2. Recurrent Neural Networks (RNN)
3. Convolutional Neural Network (CNN)
4. **Bidirectional Encoder Representations from Transformers (BERT)**

BERT

Motivation & Architecture

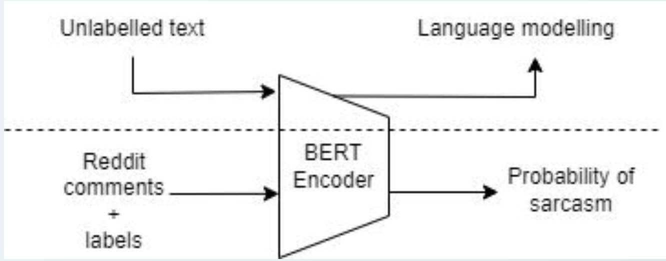
- **Contextual, bidirectional** word embeddings (vs context-free in GloVe)
- **Pre-trained transformer encoder** that can be used to to solve other problems (transfer learning)



BERT

Motivation & Architecture

- **Contextual, bidirectional** word embeddings (vs context-free in GloVe)
- **Pre-trained transformer encoder** that can be used to to solve other problems (transfer learning)



Results

Data pre processing type	Accuracy	Precision	Recall	F1 Score
No pre-processing	0.7603	0.7558	0.7704	0.7630
Removing repeating symbols	0.7642	0.7735	0.7484	0.7607
Removing repeating symbols + no contractions	0.7588	0.7457	0.7866	0.7656

Results

Data pre processing type	Accuracy	Precision	Recall	F1 Score
No pre-processing	**0.7603**	**0.7558**	**0.7704**	**0.7630**
Removing repeating symbols	0.7642	0.7735	0.7484	0.7607
Removing repeating symbols + no contractions	0.7588	0.7457	0.7866	0.7656

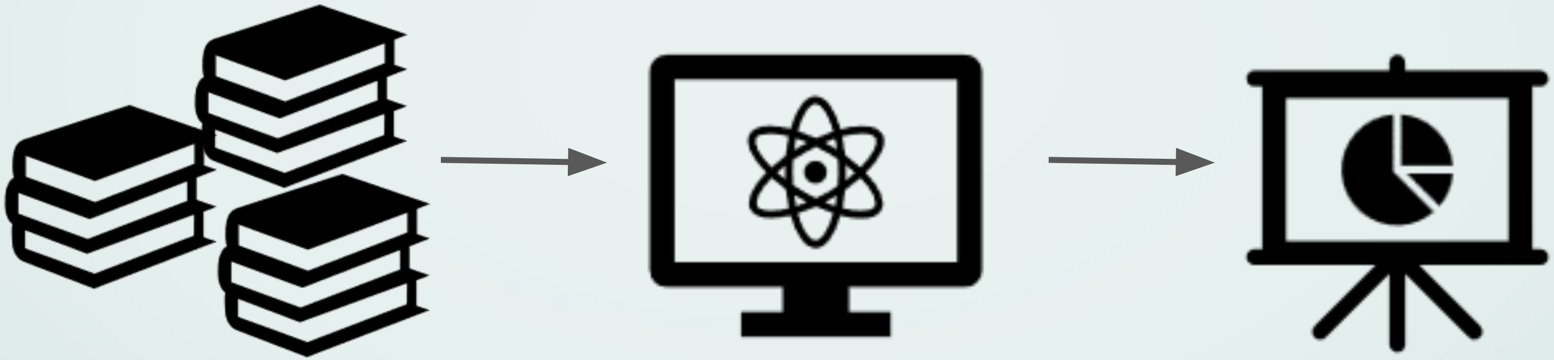
05

Results Interpretation

Micro and Macro Evaluation

Macro-Evaluation

Macro-Evaluation Approach



Test Dataset **Model** **Accuracy F1 Score**

Macro Evaluation

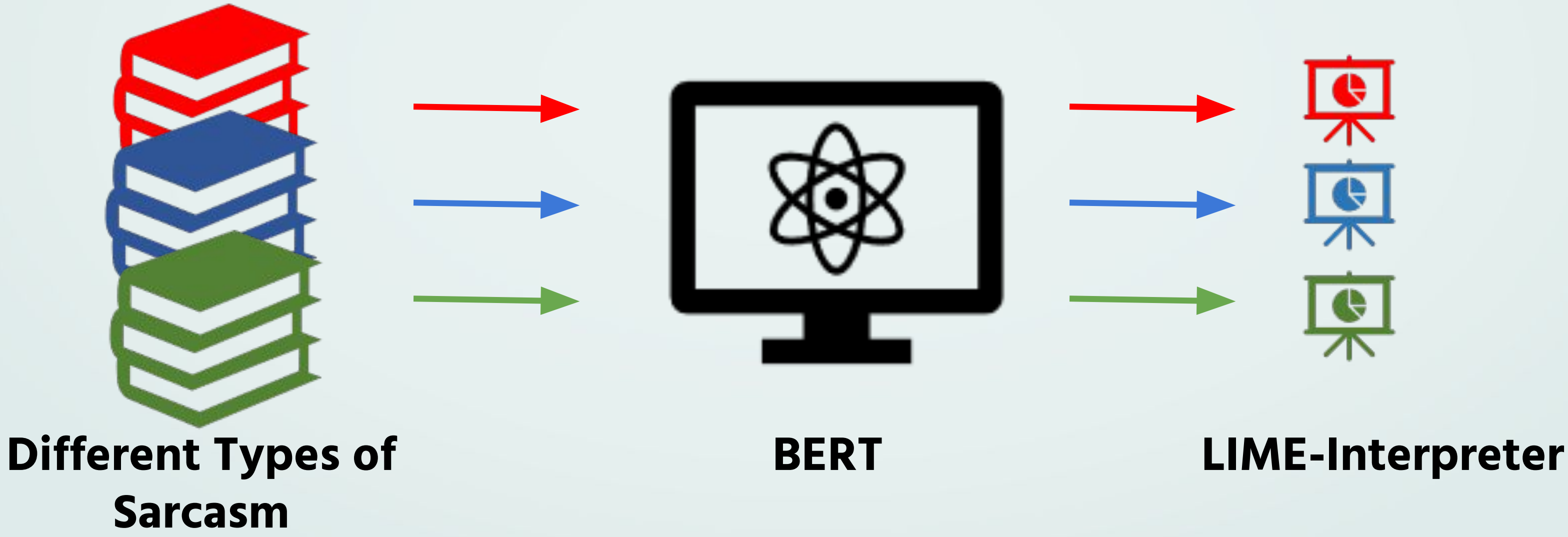
Summary

Model	Accuracy	Precision	Recall	F1 score
Logistic Regression	63.99%	0.654	0.595	0.623
RNN	71.80%	0.731	0.683	0.700
CNN	74.21 %	0.758	0.712	0.734
BERT	76.03%	0.756	0.770	0.763

Micro-Evaluation

In terms of different categories for sarcasm

Micro-Evaluation Approach



Types of Sarcasm

01

Embedded

Sarcastic Sentences that showcase **extremities within the same statement.**

E.g. If had a dollar for every **smart** thing you say. I'll be **poor**.

02

'Like'-Prefixed

Sarcastic statements that **begin with 'like' or 'as if'** to showcase a difference in intentions.

E.g. **Like** you care.

03

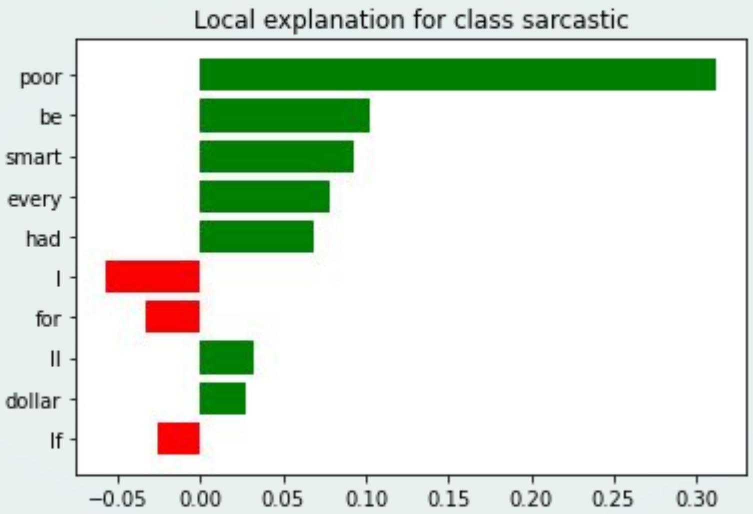
Propositional

Sarcastic statements that may **require some form of contextual knowledge** to understand.

E.g. Your plan sounds **fantastic**!

Based on Sarcasm, Pretense, and The Semantics/Pragmatics Distinction (Camp, 2012)

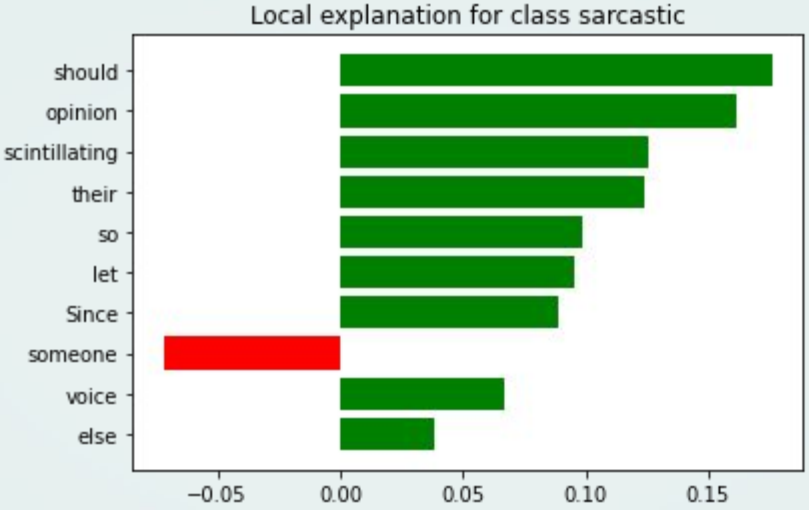
Embedded Sarcasm



If had a dollar for every **smart** thing you say. I'll be **poor**.

Prediction : Sarcastic

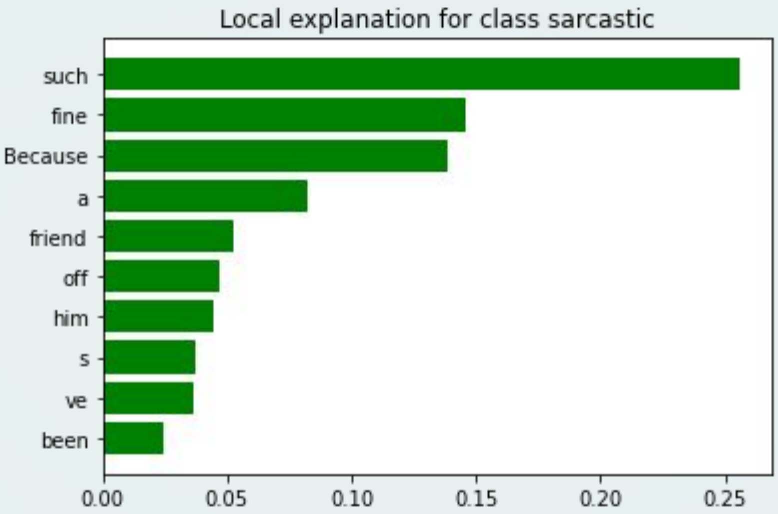
Embedded Sarcasm



Since you've already made so many **scintillating** points this evening, I think you should let someone else voice their opinion.

Prediction : Sarcastic

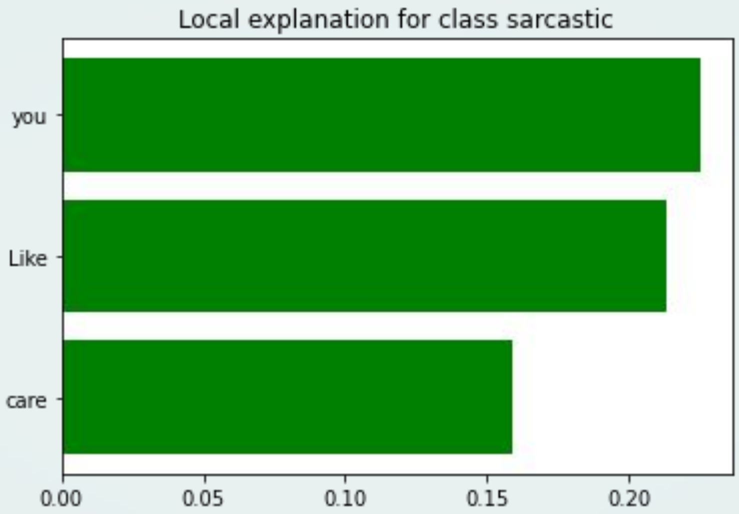
Embedded Sarcasm



Because he’s been **such** a **fine** friend, I’ve struck him off my list.

Prediction : Sarcastic

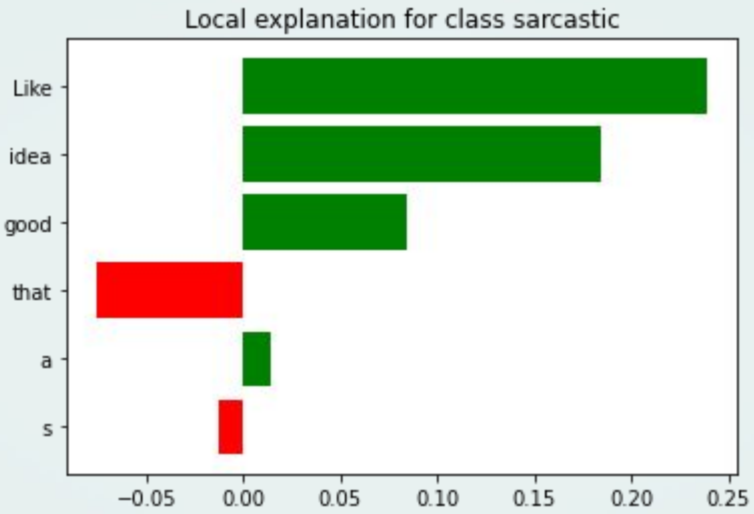
'Like'-Prefixed Sarcasm



Like you care

Prediction : Sarcastic

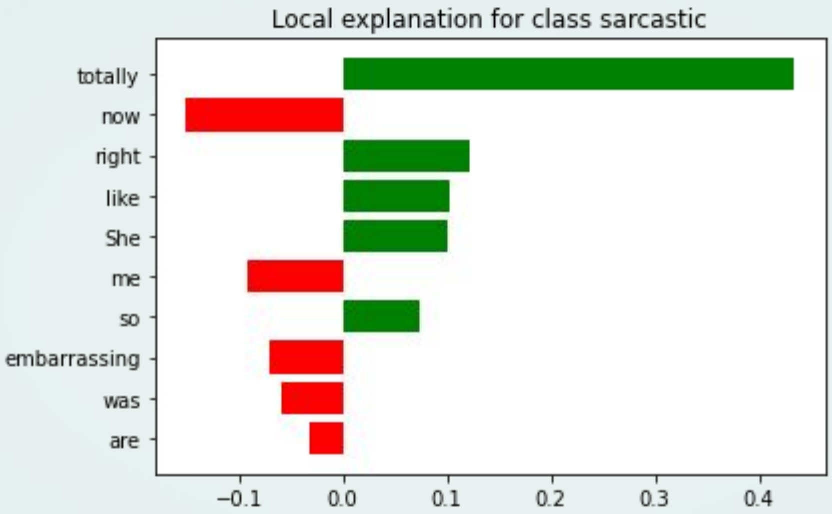
‘Like’-Prefixed Sarcasm



Like that’s a good idea.

Prediction : Sarcastic

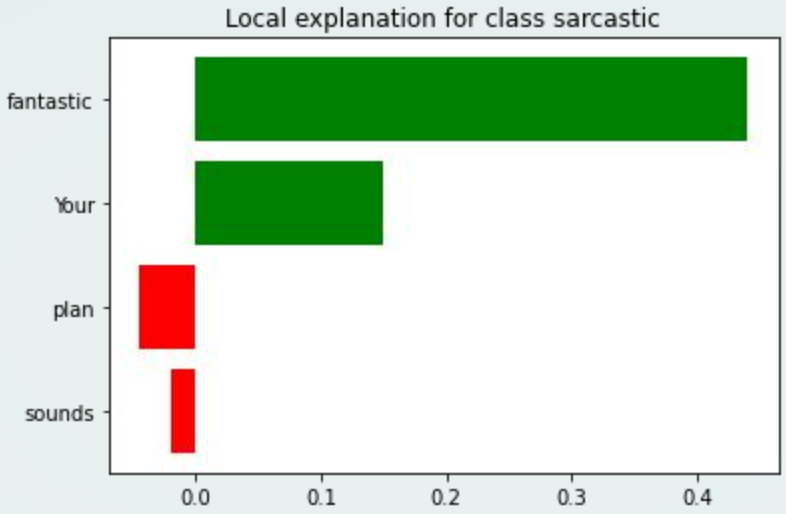
‘Like’-Prefixed Sarcasm



She was **like**, you are so totally embarrassing me right now

Prediction : Sarcastic

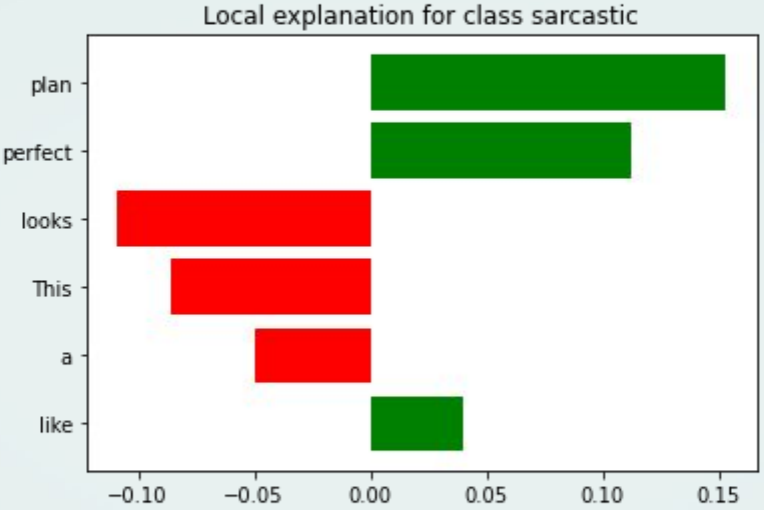
Propositional Sarcasm



Your plan sounds **fantastic**.

Prediction : Sarcastic

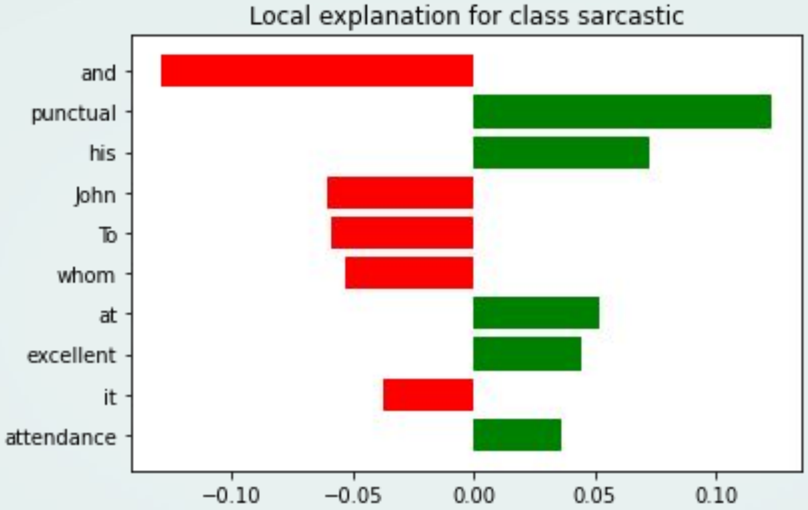
Propositional Sarcasm



This **looks** like a **perfect** plan!

Prediction : Not Sarcastic

Propositional Sarcasm



To whom it may concern: John's handwriting is excellent and his attendance at departmental events is punctual.

Prediction : Not Sarcastic

Conclusion

Model	Accuracy	Precision	Recall	F1 score
Logistic Regression	63.99%	0.654	0.595	0.623
RNN				
CNN	74.21 %	0.758	0.712	0.734
BERT	76.03%	0.756	0.770	0.763

Improvement in models

Discarding preprocessing

Preventing the loss of important information

Feature engineering

Enhancing models' ability to generalise better

Findings & Future Improvements

Finding #1

Poor LIME Interpretation prediction for **Propositional Sarcasm**

Improvement #1

Contextualization with *parent comment*

	Accuracy	Precision	Recall	F1 score
Baseline RNN	69.01%	0.633589	0.635086	0.634336
RNN with parent comment	69.92%	0.669575	0.571175	0.616473

Findings & Future Improvements

Finding #1

Poor LIME Interpretation prediction for Propositional Sarcasm

Improvement #1

Contextualization with *parent comment*

Finding #2

Prediction for **Embedded Sarcasm** can be further improved **using sentiment based features**

Improvement #2

Evaluating **positive** and **negative** words to identify **polarity shifts** in sentences

Improvement #3

Evaluating **positive** and **negative** words to identify **polarity shifts** in sentences

Sentiment-based features:

Count of:

1. Positive words
2. Negative words
3. Highly emotional positive words
4. Highly emotional negative words

Based on:
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3384025

Thank you

References

[1] Use of Punctuation in Sarcasm
https://style.mla.org/scare-quotes-origins/
[2] Literature on sarcasm detection
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3384025
[3] How sarcasm impedes sentiment analysis
https://www.researchgate.net/publication/267265686_Irony_and_Sarcasm_Corpus_Generation_and_Analysis_Using_Crowdsourcing
[4] Literature on sarcasm detection
https://nlp.stanford.edu/seminar/details/pbhattacharyya.pdf
[5] Types of sarcasm
https://www.sas.upenn.edu/~campe/Papers/Camp.Sarcasm.pdf
[6] Stacked Bidirectional LSTM Based Framework for Sarcasm Identification
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9316208