Don’t get it? Aka the sarcasm

Sarcasm Detection

The Namespace: **Tanish Sawant, Shwe Han**, Muskaan Chopra, Anshul Gupta

Motivation

"Was that sarcasm?"
"No" (sarcastically)

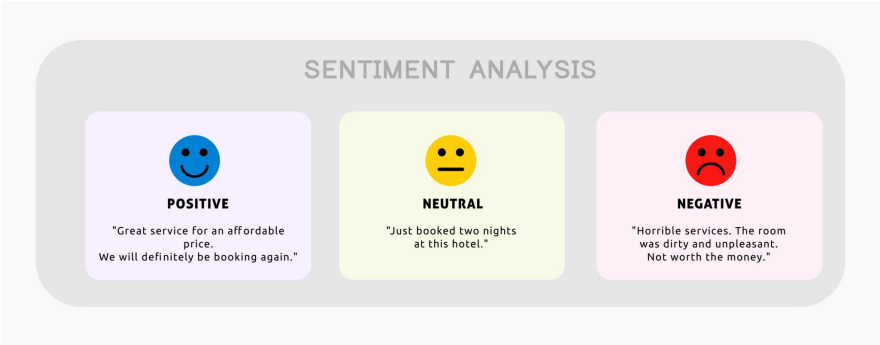
Believe it or not,

Sarcasm

- Able to catch the user's attention more than some normal text
- Adding a fun element, it sometimes opens up a way for critics not limited to certain sections of the society may it be political, social, or some other

These have even started appearing in marketing strategies of companies like Zomato through pop-up notifications or other means.

Problem Definition

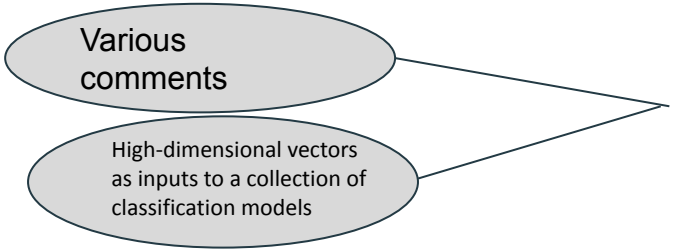


- Detecting sarcasm is not that straight forward
- Even in some settings, it is difficult to recognize by humans

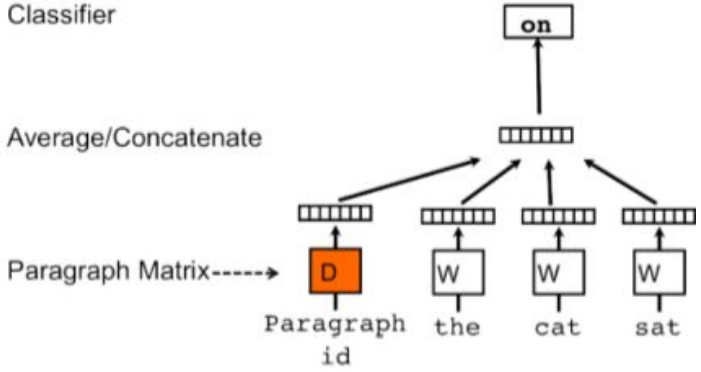
We are approaching this in three ways to detect sarcasm comments on reddit

Related Work

1. Classifying Sarcastic Comments on Reddit Using Word Embeddings



To classify comments



- compare the performance of word embedding models to traditional text-classification techniques using a bag-of-n-grams representation
- Five-fold cross-validation results on various models showed that **simple Bernoulli Naive-Bayes classifiers using unigrams and bigrams worked best**, with validation error rates as low as 30.9%.

2. Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM

- In this paper, a multi-head attention-based bidirectional long-short memory (MHA-BiLSTM) network to detect sarcastic comments in a given corpus
- extracted the most significant features and built a feature-rich SVM that outperforms these models
- experiment results revealed that a multi-head attention mechanism enhances the performance of BiLSTM, and it performs better than feature-rich SVM models.

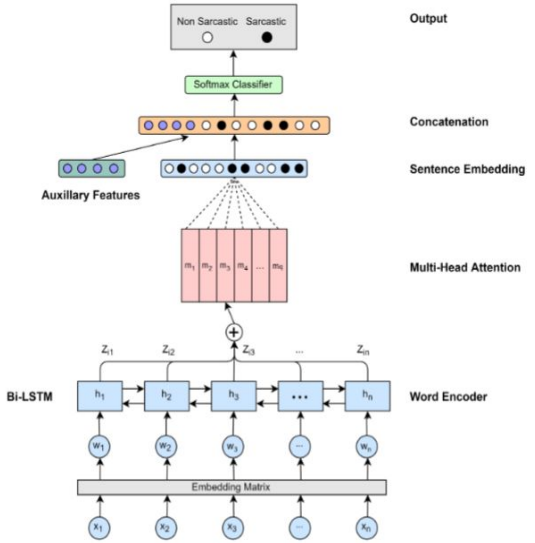


FIGURE 2. Multi-Head Self Attention Network showing the sentence embedding model combined with a fully connected and softmax layer for sarcasm detection.

TABLE 5. Precision, Recall and F-score of our approaches on datasets.

Approach	Balanced			Imbalanced		
	Precision	Recall	F-score	Precision	Recall	F-score
SVM	72.30%	75.97%	74.09%	52.14%	53.70%	52.91%
BiLSTM	69.41%	77.75%	73.34%	59.94%	43.18%	50.20%
MHA-BiLSTM-w/o-auxiliary-features	71.92%	80.02%	75.76%	59.61%	50.00%	54.38%
MHA-BiLSTM	72.63%	83.03%	***77.48%***	60.26%	53.71%	***56.79%***

Data

- This dataset contains 1.3 million Sarcastic comments from the Internet commentary website Reddit.
- Text with the tag sarcastic was Scraped from the website.
- Data has a true distribution with a true ratio of 1:100.
- The dataset originally was generated by the creators via scraping comments from Reddit (not by us :)) 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.

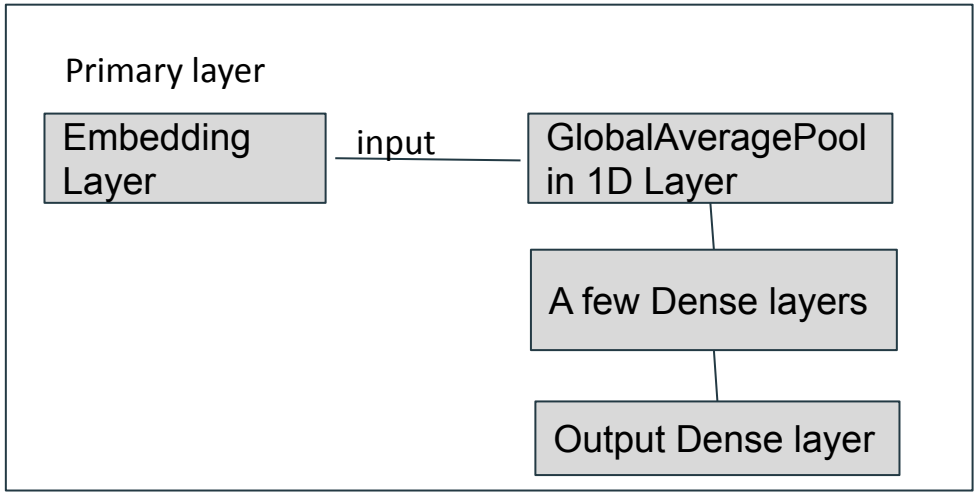
	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	NC and NH.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	Yeah, I get that argument. At this point, I'd ...
1	0	You do know west teams play against west teams...	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	The blazers and Mavericks (The wests 5 and 6 s...
2	0	They were underdogs earlier today, but since G...	Creepeth	nfl	3	3	0	2016-09	2016-09-22 21:45:37	They're favored to win.
3	0	This meme isn't funny none of the "new york ni...	icebrotha	BlackPeopleTwitter	-8	-1	-1	2016-10	2016-10-18 21:03:47	deadass don't kill my buzz
4	0	I could use one of those tools.	cush2push	MaddenUltimateTeam	6	-1	-1	2016-12	2016-12-30 17:00:13	Yep can confirm I saw the tool they use for th...

Preprocessing...

- Stemming words using PorterStemmer
- Remove unnecessary tokens and punctuations, urls, mentions, hashtags, html tags using reg
- Tokenizing sentences and padding the sequences
- Tf-IDF vectorization

Approaches

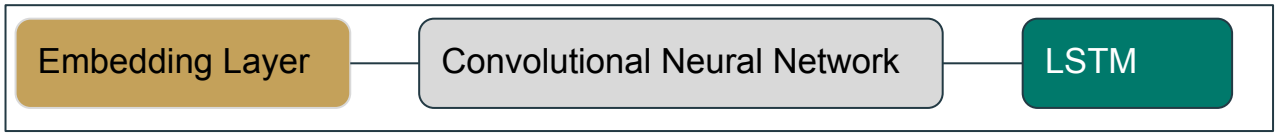
1. Simple neural network



2. Using different classification methods

logistic regression | multinomial naive bayes | XGBoost

3. Neural Network of embedding layer+ CNN + Long Short Term Memory Layer



Optimization & Results

Logistic Regression:

C value	Training Accuracy Score	Test Accuracy Score
0.001	0.53021	0.52998
0.01	0.61502	0.61432
0.1	0.65979	0.65527
1	0.68460	0.66514
5	0.70106	0.66108
10	0.70221	0.65922

MultinomialNB:

Alpha value	Training accuracy score	Test accuracy score
0	0.68991	0.652435
0.2	0.68974	0.653302
0.6	0.68914	0.654501
0.8	0.68883	0.654896
1	0.68859	0.655275

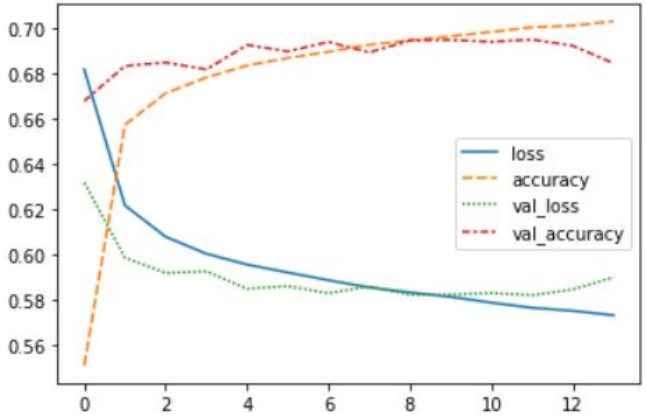
XGBoost:

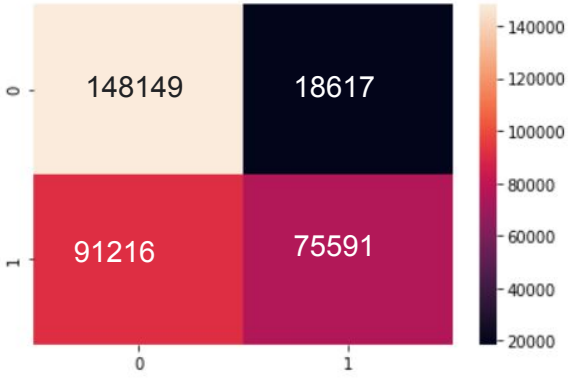
n_estimators	Training accuracy score	Test accuracy score
100	0.65674	0.64746
200	0.67160	0.65836
300	0.68160	0.66416

Simple Neural Network- Embedding Layer + dense layers

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 5)           2000005
_________________________________________________________________
dropout (Dropout)            (None, None, 5)           0
_________________________________________________________________
global_average_pooling1d (Gl (None, 5)                 0
_________________________________________________________________
dropout_1 (Dropout)          (None, 5)                 0
_________________________________________________________________
dense (Dense)                (None, 16)                96
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 17
=================================================================
Total params: 2,000,118
Trainable params: 2,000,118
Non-trainable params: 0
```

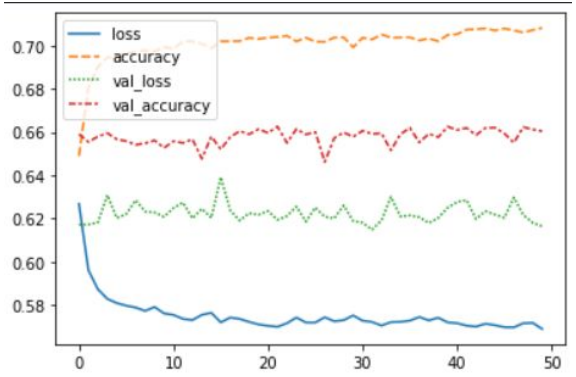
```
accuracy_score(y_pred, y_test) : 0.6783762474780632
f1_score(y_pred, y_test): 0.6026878793602121
r2_score(y_pred, y_test): -0.5059293408210332
jaccard_score(y_pred, y_test): 0.43131943855482996
```

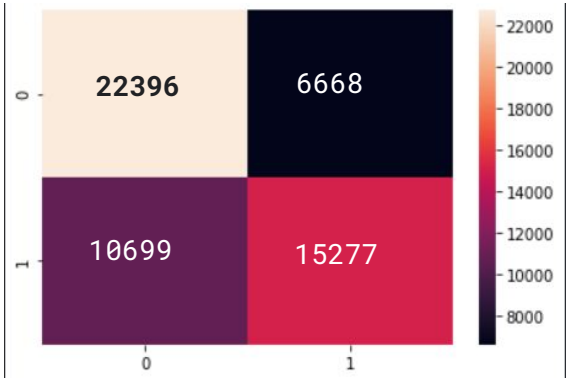




Embedding Layer + CNN + LSTM

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, None, 30)          2469090
_________________________________________________________________
conv1d_5 (Conv1D)            (None, None, 256)         38656
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, None, 256)         0
_________________________________________________________________
dropout_10 (Dropout)         (None, None, 256)         0
_________________________________________________________________
lstm_5 (LSTM)                (None, 25)                28200
_________________________________________________________________
dense_13 (Dense)             (None, 128)               3328
_________________________________________________________________
dropout_11 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_14 (Dense)             (None, 32)                4128
_________________________________________________________________
dense_15 (Dense)             (None, 1)                 33
=================================================================
Total params: 2,543,435
Trainable params: 2,543,435
Non-trainable params: 0
_________________________________________________________________
```





Comparison

	Logistic Regression (L1)	Logistic Regression (L2)	Multinomial NB	XGBoost	Single Layer NN	Embedding + CNN +LSTM
Training Accuracy	0.7010	0.7007	0.6888	0.7822	0.6267	0.6353
Test Accuracy	0.6610	0.6615	0.6548	0.6728	0.6783	0.6856
F1 score	0.5817	0.5840	0.5503	0.5757	0.6026	0.6024
Jaccard Score	0.4105	0.4124	0.4012	0.4042	0.4313	0.4310

Conclusion

Sarcasm detection is neither new nor old problem we have tackled. It is profoundly contextual and needs the speaker and the audience some shared knowledge between them. With the time that we have, we studied how the existing studies work and we tried different optimization and we found that there are some failures and achievements that we have faced.

Future Work

- Using different embedding layers (such as GloVe, BERT, etc)
  - Currently using standard embedding in keras
- Using parent comment (post) for more contextual understanding

	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	NC and NH.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	Yeah, I get that argument. At this point, I'd ...
1	0	You do know west teams play against west	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	The blazers and Mavericks (The wests 5 and 6 s...

- Applying this model to detect contradiction (contradiction and entailment)