Sarcasm Detection in Amazon Product Reviews

Based on the paper “A Deeper Look Into Sarcastic Tweets Using Deep Convolutional Neural Networks’
By Poria et. al.

ABHIJEET BOROLE – 2017ABPS0369P

ABHINAV TULI – 2017A7PS0048P

SANCHIT AHUJA – 2017A3PS0216P

Overview

Goal: To accurately detect sarcasm in Amazon product reviews.

- Sentiment shifting is prevalent in sarcasm-related communication.
- People with different personality types tend to express sarcasm in different ways.
- We thus trained separate **CNNs** to extract Sentiment and Personality features.
- The extracted personality and sentiment features are concatenated and fed to a **Support Vector Machine** which classified reviews as Sarcastic/Not Sarcastic.

Word Embeddings

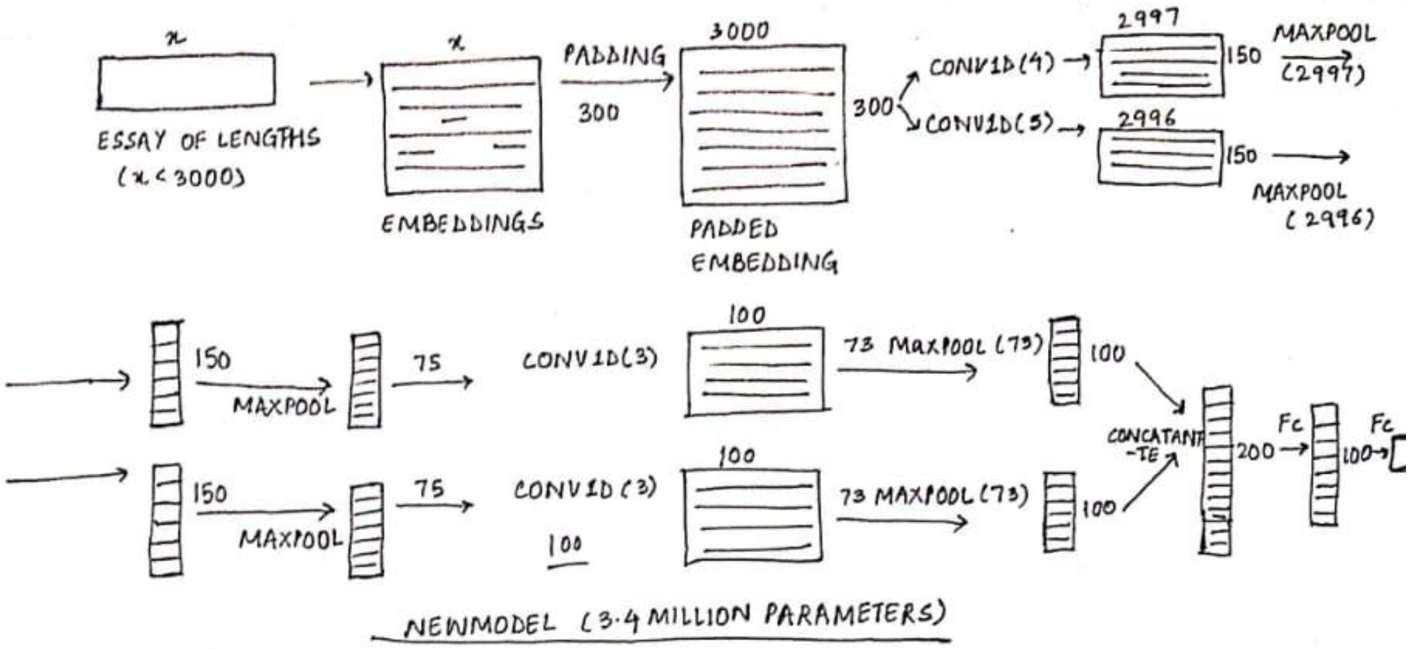
- A **word embedding** is a learned representation for text where words with the same meaning have a similar representation.
- All of our CNNs take word embeddings of text as input.
- We have used **non-static Stanford GloVe vectors** that uses a continuous bag-of-words architecture.
- Non-static representations are necessary as the GloVe vectors are not equipped to handle sarcasm.
- The presence of informal language is handled by non-static representations as well.

Sentiment Extraction using CNN

- **Sentiment analysis** is the interpretation and classification of emotions (positive and negative) within text data
- For training, we used the **IMDB dataset** consisting of 50000 highly polar movie reviews classified as either Positive or Negative.
- The train, validation and test split was **17500, 7500 and 25000** respectively.
- We experimented with a couple of CNN architectures, each having its own pros and cons.
- Once the model was trained, we could extract the sentiment features from the first fully connected layer.

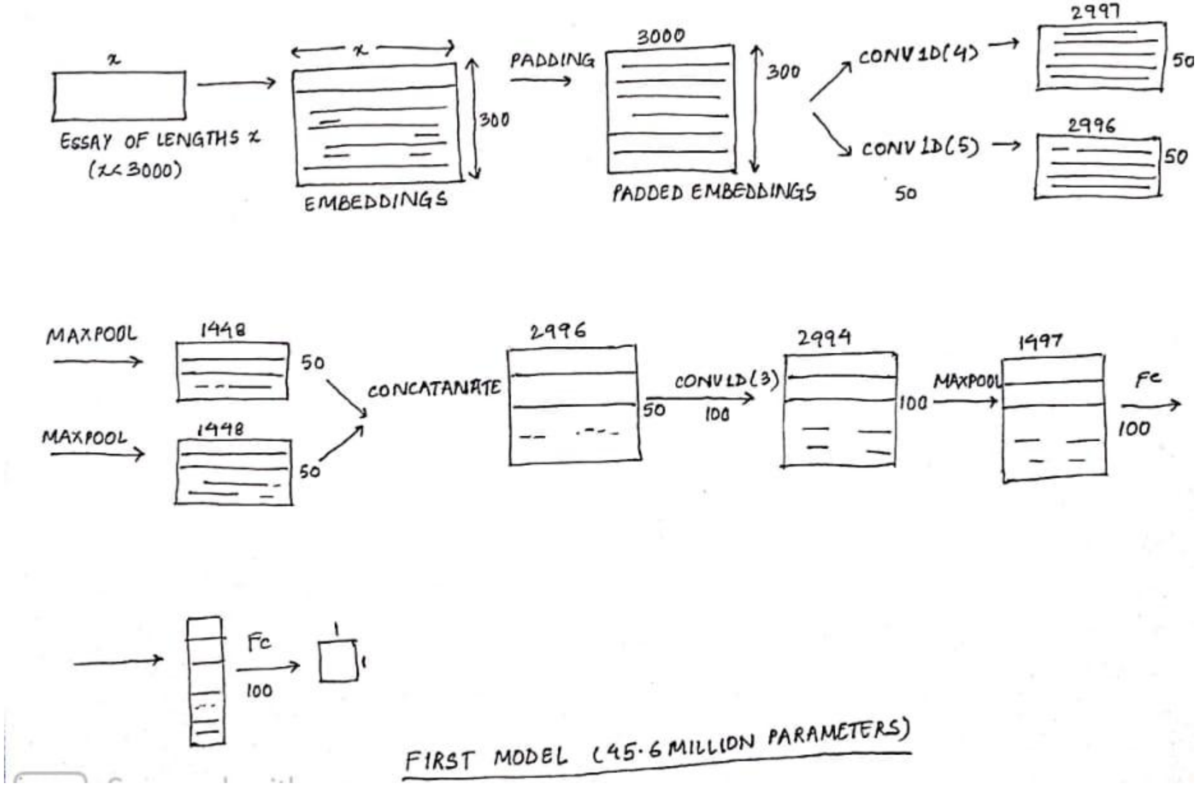
Architecture of Proposed Sentiment Model in Paper

- This model has just 3.4 Million Parameters and is very easy to train.



Architecture of Modified Model

- This model has just 45.6 Millions parameters and is slower to train, however it gives us a better accuracy.



Sentiment Results

- For the final model we went ahead with the model proposed in paper as it was much easier to train and also closer to our initial object.

Model	Test Loss	Test Accuracy
Proposed in paper	0.351	85.05%
Our modified	0.279	88.74%

Our Modified

```
Epoch: 01 | Epoch Time: 0m 47s
       Train Loss: 0.601 | Train Acc: 67.61%
        Val. Loss: 0.304 |  Val. Acc: 87.55%
Epoch: 02 | Epoch Time: 0m 47s
       Train Loss: 0.237 | Train Acc: 90.75%
        Val. Loss: 0.263 |  Val. Acc: 89.57%
Epoch: 03 | Epoch Time: 0m 48s
       Train Loss: 0.094 | Train Acc: 96.74%
        Val. Loss: 0.324 |  Val. Acc: 89.38%
Epoch: 04 | Epoch Time: 0m 48s
       Train Loss: 0.023 | Train Acc: 99.32%
        Val. Loss: 0.578 |  Val. Acc: 86.52%
Epoch: 05 | Epoch Time: 0m 48s
       Train Loss: 0.015 | Train Acc: 99.57%
        Val. Loss: 0.579 |  Val. Acc: 88.40%
```

Proposed In Paper

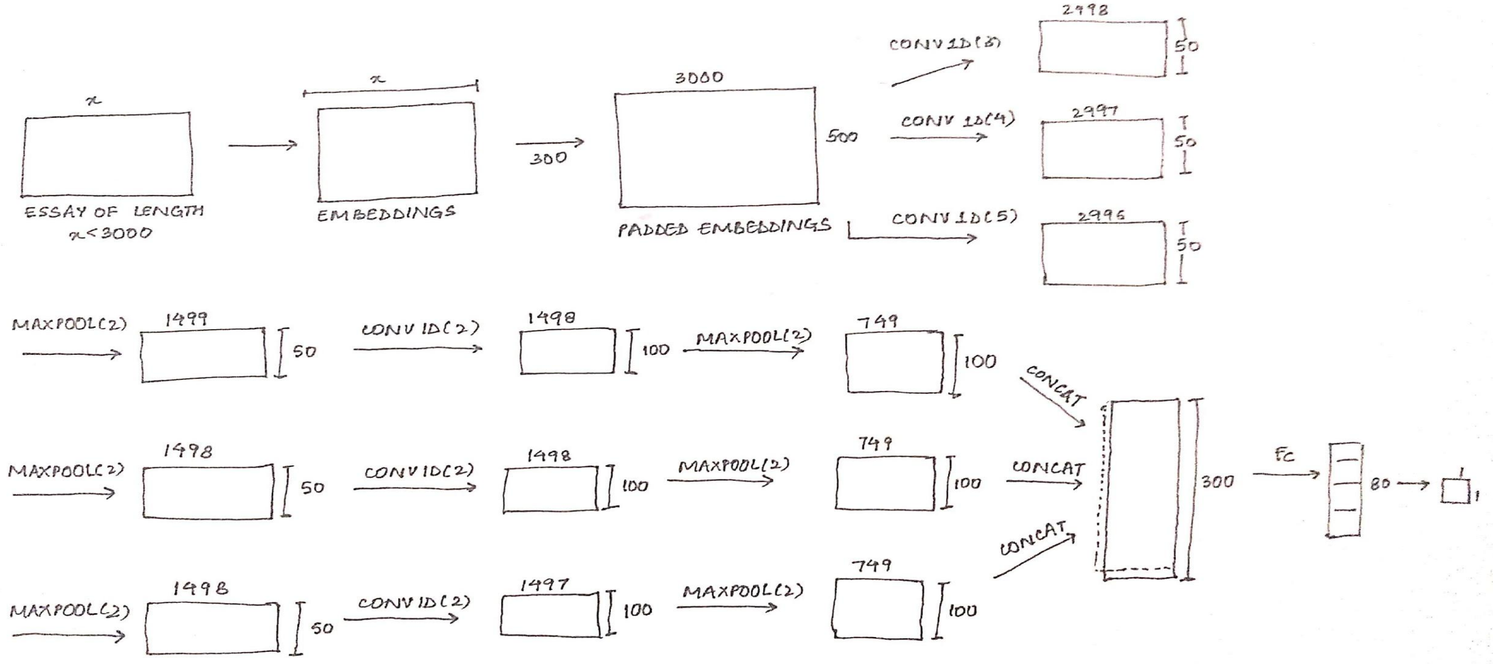
```
Epoch: 01 | Epoch Time: 1m 19s
       Train Loss: 0.589 | Train Acc: 65.21%
        Val. Loss: 0.409 |  Val. Acc: 81.55%
Epoch: 02 | Epoch Time: 1m 18s
       Train Loss: 0.345 | Train Acc: 85.33%
        Val. Loss: 0.357 |  Val. Acc: 84.04%
Epoch: 03 | Epoch Time: 1m 18s
       Train Loss: 0.219 | Train Acc: 91.59%
        Val. Loss: 0.352 |  Val. Acc: 85.33%
Epoch: 04 | Epoch Time: 1m 18s
       Train Loss: 0.116 | Train Acc: 95.87%
        Val. Loss: 0.458 |  Val. Acc: 84.39%
Epoch: 05 | Epoch Time: 1m 18s
       Train Loss: 0.049 | Train Acc: 98.26%
        Val. Loss: 0.613 |  Val. Acc: 84.64%
```

Personality Extraction using CNN

- The dataset that we used to classify personality is called OCEAN dataset. It consists of five personalities:
  - OPN - [O] Openness to experience. (inventive/curious vs. consistent/cautious)
  - CON - [C] Conscientiousness. (efficient/organized vs. easy-going/careless)
  - EXT - [E] Extroversion. (outgoing/energetic vs. solitary/reserved)
  - AGR - [A] Agreeableness. (friendly/compassionate vs. challenging/detached)
  - NER- [N] Neuroticism. (sensitive/nervous vs. secure/confident)

- We had run couple of experiments dabbling with sending the complete text as well as chunking the text into sentences and then sending them to the network
- Chunking the text into sentences performed slightly better than sending the complete text
- Experimented with different sizes of fully connected layers at the end and tabulated the results
- We experimented on the Openness personality and extrapolated the results to the other personalities

Architecture of Personality Model



Personality Experimentation

Complete Text

FC Layer Size	Train acc.	Val acc.
70	92.05	58.32
80	99.85	62.30
90	98.19	58.50
100	99.95	58.84
110	99.55	60.48
120	89.06	59.45
130	99.50	58.47
140	99.75	58.27
150	99.95	62.15

Text chunked into sentences

FC Layer Size	Train acc.	Val acc.
80	86.63	62.41
90	92.04	59.91
100	99.19	60.20
110	98.89	60.37
120	96.10	60.94
130	94.90	61.00
140	99.75	60.92
150	99.95	62.04

- The best result we obtained was with chunked text with 80 fully connected layers

Personality Results

- The best results we obtained for each personality:

Personality	Val acc. (Using complete text)	Val acc.(Using chunks)
OPN	57.63	58.41
NEU	56.75	57.55
EXT	55.65	55.96
CON	52.89	52.19
AGR	53.28	52.65

Sarcasm Detection

- We concatenated the features extracted from the personality and sentiment models
- We trained the Sarcasm detection classifier on the imbalanced Amazon Reviews dataset(1254 Reviews) after extracting their features as discussed above
- We explored several options:
    - We ran several classifiers such as MLP and SVM
    - We experimented with the features, which included running the classifier on Sentiment features only
- We ran a Grid-Search to find the optimal hyperparameters for the SVM

Results

SVM Using Sentiment and Personality

```
Confusion Matrix:
[[73  5]
 [10 38]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.94      0.91        78
           1       0.88      0.79      0.84        48

    accuracy                           0.88       126
   macro avg       0.88      0.86      0.87       126
weighted avg       0.88      0.88      0.88       126

Accuracy: 0.8809523809523809
```

MLP Using Sentiment and Personality

```
Confusion Matrix:
[[69  9]
 [16 32]]
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.88      0.85        78
           1       0.78      0.67      0.72        48

    accuracy                           0.80       126
   macro avg       0.80      0.78      0.78       126
weighted avg       0.80      0.80      0.80       126

Accuracy: 0.8015873015873016
```

SVM using Sentiment only

```
Confusion Matrix:
[[72  6]
 [11 37]]
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.92      0.89        78
           1       0.86      0.77      0.81        48

    accuracy                           0.87       126
   macro avg       0.86      0.85      0.85       126
weighted avg       0.86      0.87      0.86       126

Accuracy: 0.8650793650793651
```

As evident, we obtained the best results with an SVM Using Sentiment and Personality features. The accuracy **(weighted F1-score)** of the same was **88%**.

THANK YOU