SARSASM TARGET DETECTION

Guided By:

Jasabanta Patro

Presented By:

Mayank Jain

Pranjal Doshi

Yogesh Shelke

Rachit Agarwal

PROBLEM STATEMENT

- Sarcasm Target Identification

1. Extracting Sarcasm target – Entity or situation being ridiculed in a sarcastic text.
2. Input: Sarcastic text
3. Output:
   - Subset of words in the sentence that point to the sarcasm target
   - Fall-back label ‘Outside’
4. What’s the need?
   - Challenge to sentiment analysis.
   - Example: *“My cell phone has an awesome battery that lasts 20 minutes.”*

DATASET

- https://github.com/Pranav-Goel/Sarcasm-Target-Detection
- Manually annotated
  - Book Snippets (224)

	Snippets
Count	224
Average #words	28.47
Vocabulary	1710
Total words	6377
Average length of sarcasm target	1.6
Average polarity of sarcasm target	0.0087
Average polarity of portion apart from sarcasm target	0.027

Baseline Results

- Dice Score : 32.68
- Exact Match : 7.01%

OUR APPROACH

- 1) KNN
- 2) DECISION TREE
- 3) RANDOM FOREST
- 4) CNN
- 5) BI_LSTM
- 6) POSTAG BASED STATISTICAL CLASSIFIER
- 7) RULE BASED APPROACH

DataSet at a glance

- #Examples : 224
- Average no of words in a sentence : 30
- Min sentence length : 6 words
- Max sentence length : 76 words
- Average sentence length : 30 words
- All sentences were unilingual (English)

- To make every sentence of equal size null padding was done.

- We considered this as a sequence labelling problem.
- So our problem is basically a 77 class classification problem.

  (76 words + 1(outside))

You	really	are	a	ray	of	sunshine,	aren't	you?
1	0	0	0	0	0	0	0	1

WORD EMBEDDINGS

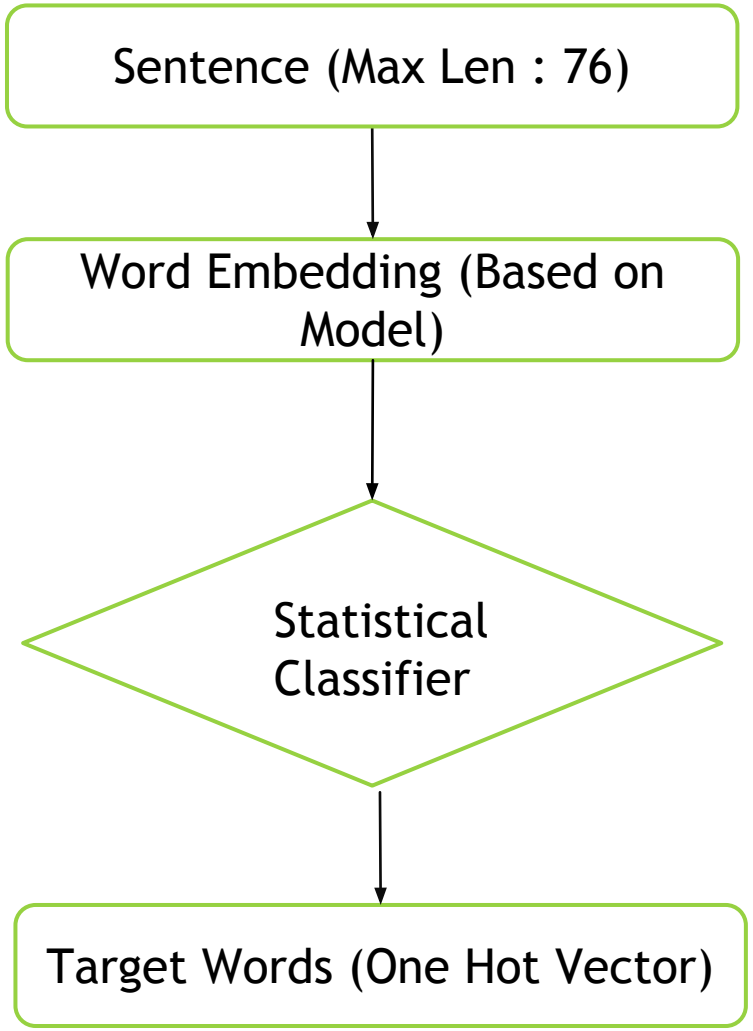
- We started with various word embedding like:
- Here are the ones which gave us best results:
  - 1) Word2vec (vector size: 100)
  - 2) GLoVe (vector size: 10)
  - 3) Fasttext (vector size: 50)

These embedding were passed on to various models.

Evaluation metric

- We considered accuracy as metric to evaluate our models.
- 2 cases:
  - 1) partial match: if our model returns one of the target word as output, we considered that example as ”correctly labeled”.
  - 2) Complete Match; if our model returns all the target words as output, then only that example is considered as ‘correctly labeled”.

Statistical Classifier Model



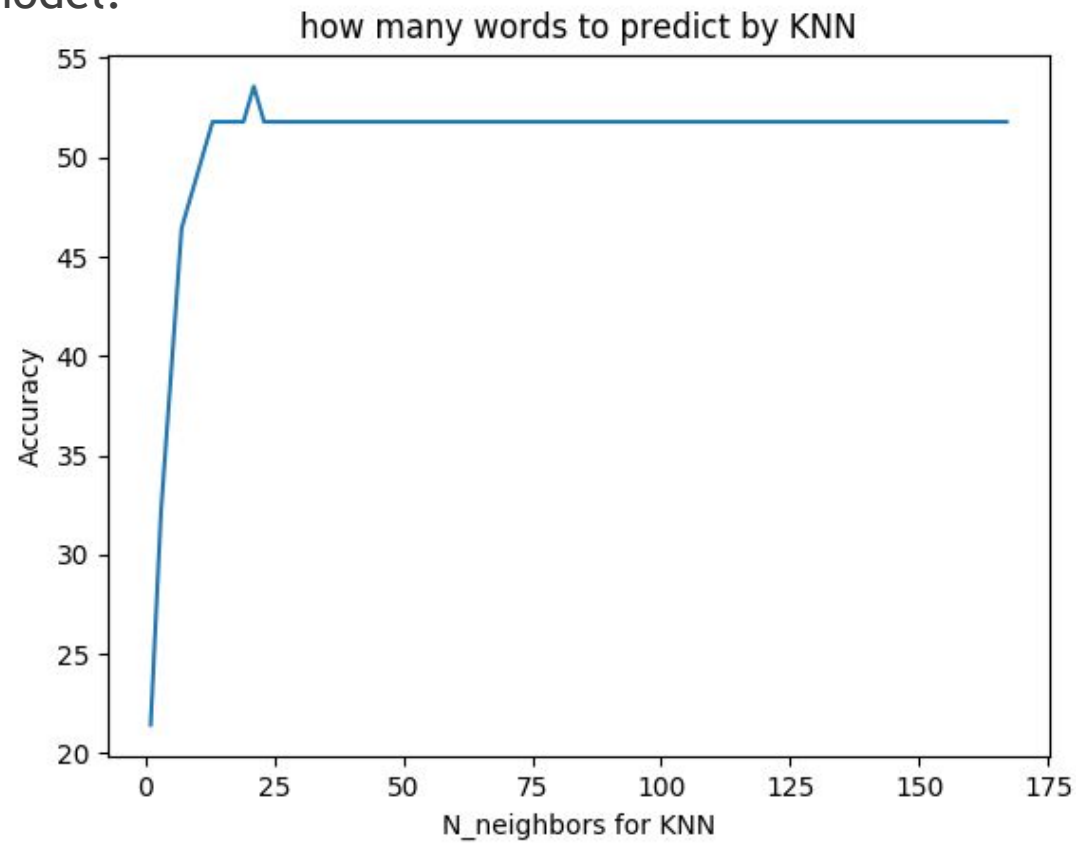
KNN

- Word embedding : Fasttext
- Vector length : 50

- Accuracy:
- Partial Match : 7.14 %
- Dice Score : 4.39
- Complete Match : 0.00 %

Intuition From Previous Approach

- Our model should learn how many words to predict as target words.
- We used KNN model for predicting that for our next model.
- Accuracy of model was 55% at n_neighbors = 21.



Decision tree

- Embedding used: GLoVe(vector length: 10)

- Accuracy:
- Partial Match : 30.35 %
- Dice Score : 16.52
- Complete Match : 7.14 %

Random Forest

- As Decision tree gave good results, we expected Random Forest to give good results as well.
- 2 level of learning was used.
- One to learn how many words to return(k).
- And other to predict and return top k target words.

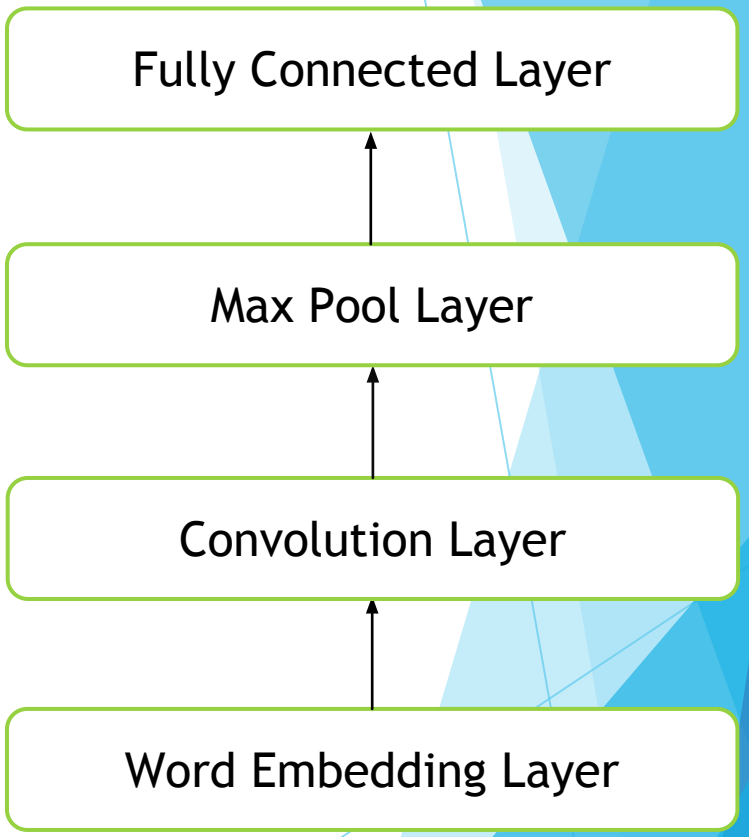
- Embedding used: GLoVe(vector length: 10)

- Accuracy:
- Partial Match : 41.78 %
- Dice Score : 25.30
- Complete Match : 19.18 %

Deep Learning approaches.

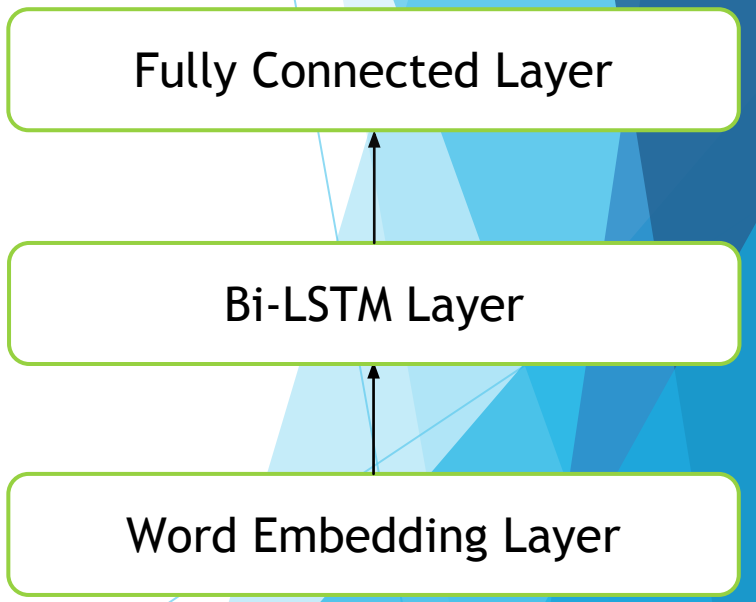
CNN

- Word Embedding Layer : Word2Vec(vector length : 100)
- Loss Function : Softmax
- Stride : (1,1)
- Accuracy:
  - Partial Match : 25.14 %
  - Dice Score : 9.53 %
  - Complete Match : 1.7 %



Bi-LSTM

- Word Embedding Layer: Fasttext(vector length : 100)
- target words (y_{ik}) = $argmax_k(softmax(y_{pred}))$
- Accuracy:
  - Partial Match : 32.14%
  - Dice Score : 24.17
  - Complete Match : 20.83%



Intuition from previous approaches

- As the task is not easy, Using plain ML and DL techniques wont help.
- We need to mine some “FEATURES” out of the data to get better accuracies.
- We tried 2 approach:
- 1) Rule based approach
- 2) Statistical Approach

Statistical Approach to mine features

- Intuition: Given a sequence of POS-Tags, find a trend about which type of POS-Tags are generally returned as target tags.

- We built a statistical model to learn this trend.

Rule Based Approach to mine features

- By mining we found some rules which can be used to predict sarcasm target.
- Rules
  - Rule 1: Named Entity are likely to be candidate target.
  - Rule 2: As mentioned by Riloff et al 2013, Sentence with positive sentiment have negative sentiment verb. Then that verb can be candidate target.
    - E.g. I love being ignored.
  - Rule 3 : Subject in an interrogative sentence is likely to be candidate target.
  - Rule 4 : As mentioned by shamay et al 2005, Pronouns followed by Pronominal adjectives are likely to be candidate targets.

Merging the results

- Results :
- Statistical approach and Bi-LSTM
  - Dice Score (OR) : 35.4
  - Dice Score (AND): 25.19

Future Works

- Semantic Role Labeling
  - Assigns labels to words or phrases in a sentence that indicate their Semantic role in the sentence.
  - Describe the semantic relation between the arguments of the verb and the situation described by the verb.
    - **The boy** threw **the red ball** to **the girl**
    - **The boy** – the participant responsible for the action, the “doer”
    - **the red ball** –the affected entity, “undergoer”
    - **the girl** – endpoint in a change of location
- **Dependency Parsing** is the task of recognizing a sentence and assigning a syntactic structure to it

REFERENCES

- Sarcasm Target Identification: Dataset and An Introductory Approach Aditya Joshi et al. 2018 http://www.lrec-conf.org/proceedings/lrec2018/pdf/583.pdf

- Sarcasm as contrast between a positive sentiment and negative situation. In EMNLP, volume 13, pages 704–714 : Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013)