Sarcasm Detection in Tweets

Mahesh Tolani, Shashikant
MT2017064, MT2017104

[1]International Institute of Information Technology

mahesh.tolani@iiitb.org, shashikant.chaudhary@iiitb.org

Abstract. *This paper tries to detect sarcasm in sentences extracted from twitter. We are using data-set from SemEval-2018 Task 3. We have tried to use word embeddings from the Stanford GloVe model trained on twitter data-set(containing 2B tweets). We try to build different sets of features from these embeddings. Then we train machine learning models on these sets separately and see what results we get. In the end we combine the the features that give best results and evaluate the model using different metrics like accuracy, precision, recall, f1-score and confusion matrix.*

1. Introduction

Sarcasm is a form of verbal irony that is intended to express contempt or ridicule. Sarcasm has a negative implied sentiment, but may not have a negative surface sentiment. A sarcastic sentence may carry positive surface sentiment (for example, 'Visiting dentists is so much fun!'), negative surface sentiment (for example, 'His performance in Olympics has been terrible anyway' as a response to the criticism of an Olympic medalist) or no surface sentiment (for example, the idiomatic expression 'and I am the queen of England' is used to express sarcasm).

In this paper, we try to use word embeddings to capture context incompatibility in the absence of sentiment words. The intuition is that word vector-based similarity/discordance is indicative of semantic similarity which in turn is a handle for context incompatibility.The way we use these similarities and dissimilarities is that we take the scores we get from the word2vec model and then use them as features. There are many word2vec models available but we have used the Stanford GloVe(Global Vectors) model. Not many attempts have been made in detecting sarcasm using word embeddings. So, basically what the paper tries to do is to answer the question :

Can we use word embeddings to generate relevant features that can be used to detect sarcasm in sentences?

2. Motivation

So, why word embeddings? We thought out that the similarity scores returned by the wrod2vec model between 2 words can be used to identify the change of context or the context incompatibility. Take for example :

similarity(man, woman) = 0.66136
similarity(sea, fire) = 0.37159

As it can be seen that when man and woman appear in a sentence it will be a normal sentence only most of the times but when fire and sea appear in a sentence it is most unlikely and hence points towards sarcasm.Hence, we propose features based on similarity scores between word embeddings of words in a sentence. In general, we wish to capture the most similar and most dissimilar word pairs in the sentence, and use their scores as features for sarcasm detection.

3. Features Based on Word-Embeddings

3.1. Mean Vectors

For each sentence we take the mean of word2vec vectors of all the words and keep this vector as a set of features. This is the first set of features that we take to train the model.

3.2. Using Windowing approach

Here, what we try to do is take a sentence and start traversing it window by window. We take the first word as one part and rest of the sentence as other part. We take the mean vector of rest of sentence and then find the Euclidean distance between the first word's vector and the mean vector. After that we include first two words together and take their mean vector and the mean vector of rest of the part and again take the euclidean distance between them. In this way we keep increasing window size from one side and decrease the size from other side. In the end 4, we take maximum and minimum euclidean distances in a sentence and take them as features.

Motivation behind this is that we wanted to detect the change in context in the sentence when it occurs. After getting features, we trained the model on these features.

3.3. Using Similarity Measures of words

Here, we use the similarity measures between words returned by the GloVe word2vec model. For every sentence we take the best and second best similarities and the worst and second worst similarities and then build a set of these four numbers as features.

4. Experimental Setup

We used data-set from SemEval-2018 Task 3. The data-set has 3817 labelled tweets. The tweets are labelled as 0 and 1 for non-sarcastic and sarcastic respectively. We do data preprocessing for the same. The preprocessing steps done are :

- Remove URLs
- Remove usernames (like @username)
- Remove hashtags
- Remove special characters

We used Support Vector Machine (with rbf kernel) and Logistic Regression for classification. Also, we used following combinations of features for training the models :

- Mean Vectors (M)
- Features from Window Approach (W)
- Similarity based features (S)
- Mean Vectors + Similarity Based features (M + S)

5. Results :

The following table shows the results that we got after testing our model on the preprocessed data.

Features	Accuracy	Precission	Recall	F1-score
M	0.6151	0.6466	0.5297	0.5823
W	0.5388	0.5536	0.4536	0.4987
S	0.5274	0.5200	0.8393	0.6422
M + S	0.6740	0.6711	0.6586	0.6648

6. Conclusion

In this paper we have shown the benefits of word embeddings in for sarcasm detection. We experimented with different features and their combinations and finally came up with mean vector features and similarity based features since they helped the most in sarcasm detection. We have just used a couple of features based on word2vec embeddings. Much more work can be done in the area, we have just touched upon the surface.

7. References

- Aditya Joshi,Vaibhav Pal et.al. [Automatic Sarcasm Detection: A Survey]
- data-set [https://competitions.codalab.org/competitions/17468]
- Aditya Joshi,Vaibhav Pal et.al. [Are Word Embedding-based Features Useful for Sarcasm Detection?]
- GloVe Model [https://nlp.stanford.edu/projects/glove/]