Detecting Sarcasm in Text

An Obvious Solution to a Trivial Problem

Chun-Che Peng, Mohammad Lakis, Jan Wei Pan

Stanford ENGINEERING
Computer Science

Abstract

Sarcasm detection in writing is challenging in part due to the lack of intonation and facial expressions. Nonetheless, the human comprehension system can often spot a sarcastic sentiment and reason about what makes it so. We attempt to design a machine learning algorithm for sarcasm detection in text by leveraging existing work and improving upon it. By analyzing the strengths and weaknesses of the baseline model, we strive to develop one that will achieve better results.

Introduction

Motivation

- Sarcasm detection is crucial in natural language studies
- Accuracy and robustness of NLP results are often affected by untruthful sentiments that are of sarcastic nature
- Often left untreated
- One of the many challenges in artificial intelligence

Statement of Problem

- In most cases, to understand sarcasm, one has to know the context of the sentence.
- Analyzing single sentences without context.
- Further limit problem by seeking to improve the existing model designed by Mathieu Cliché of www.thesarcasmdetector.com
- Used Twitter data as data points (positive points are denoted by #sarcasm)

Baseline Model

Description

- Uses a support vector machine (SVM) as implemented by the LinearSVC function from the Python library scikitlearn
- Uses 0.1 as penalty parameter C in SVM
- Tweets were collected over a span of several months in 2014
- Data sanitized of all hashtags, non-ASCII characters and http links
- Each tweet tokenized, stemmed, and un-capitalized

Feature Engineering

- *N-grams*: unigrams and bigrams
- *Sentiments*: positive and negative sentiment scores for each part of a tweet (divided into two and three equal parts)
- *Parts of Speech*: # of each type of parts of speech in each tweet
- *Capitalizations*: binary flag to indicate if tweet has at least 4 words starting with a capital
- *Topics*: collection of topics using Latent Dirichlet Allocation

Positive Examples

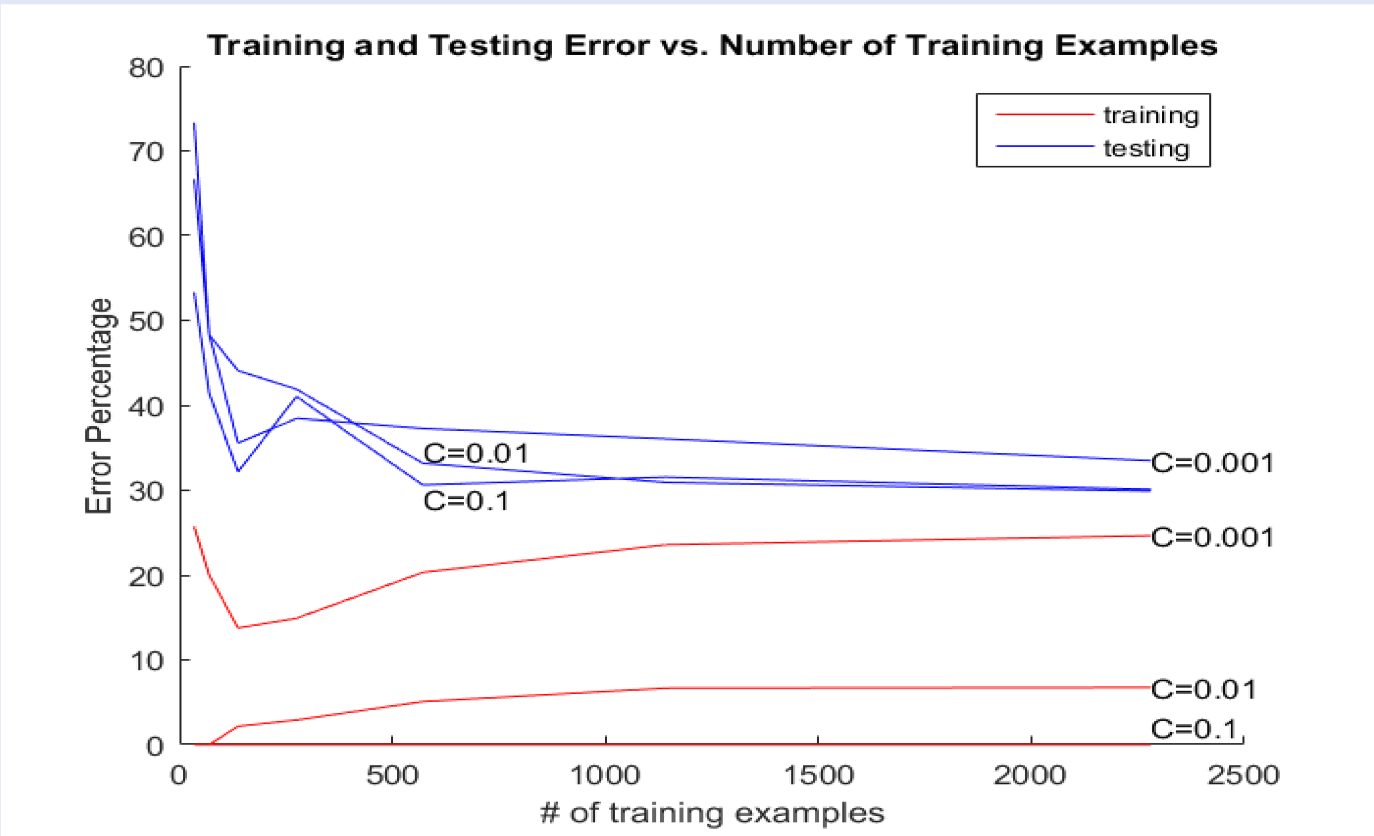
```
I just love spending this Saturday night studying...
When I grow up, I want to marry a man who spends his every spare moment in front of his computer- *Said Nobody ever*
I'm sure he'll never leave for a better job when that opens up
It's always fun to wait for a doctors appointment. I guess it could be worse, I could be waiting on a dentist
Clearly I am of the greatest importance.
```

Error Analysis

- Initial analysis of baseline model show that the testing error far exceeds training error
- Attempted different values of C
- Still obtained large gaps between testing and training error
- Suggests that baseline model suffers from high variance

Areas for Improvement

- Reduce the dimension of feature space and only use relevant features
- Sentiment analysis seemed arbitrary
- Data might not be linearly separable; try using a Gaussian kernel instead of linear
- Try using one class SVM



Model Investigations and Improvements

Naïve Bayes

- Used frequencies/Bernoulli to estimate prior and posterior probabilities from 117,879-sentences dataset.
- Performed 70/ 30 training/ validation ratio in testing dataset
- Utilized Confusion Matrix, Precision, Recall and *F*1 score to measure the performance:

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{TP + False Positive (FP)}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP + False Negative (FN)}}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Confusion Matrix} = \begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix} = \begin{bmatrix} 25180 & 60 \\ 19141 & 6137 \end{bmatrix}$$

Performance Metric				
	Precision	Recall	F1-score	Support
Non-Sarcastic Sentence	57%	100%	72%	25240
Sarcastic Sentence	99%	24%	39%	25278
Avg / total	78%	62%	56%	50518

One Class SVM

$$\min_{w,\,\xi_i,\,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho$$

subject to:

$$(w \cdot \phi(x_i)) \geq \rho - \xi_i \quad \text{for all } i = 1, \ldots, n$$
$$\xi_i \geq 0 \quad \text{for all } i = 1, \ldots, n$$

decision function $f(x) = \operatorname{sgn}((w \cdot \phi(x_i)) - \rho)$

- Twitter data was noisy, especially the sarcastic ones
- But the non-sarcastic data was clear and had little noise
- Decided to utilize one class SVM as a novelty detection algorithm to detect sarcasm, based on observing non-sarcastic data
- Used OneClassSVM function from the Python library scikitlearn on only unigrams and bigrams

Accuracy				
nu	gamma	total	negative	positive
0.1	0.1	50%	90%	10%
0.01	0.01	50%	99%	0.20%
0.5	0.5	51%	36%	67%

Feature Studies Using a Gaussian Kernel

- A linear kernel was used in the baseline model but it's likely the data isn't linearly separable
- Numerous trials revealed best results for dataset when using parameters C=0.6 and γ=1
- Re-ran model without sentiment analysis scores using a Gaussian kernel
- Obtained a testing accuracy of 82.2% but a training error of over 99%
- Shows signs of the same high variance error as the baseline model but the use of a Gaussian kernel shows potential for being better than the baseline
- However it appeared that the baseline approach achieved better results for smaller sizes of the dataset

Conclusions

- Poorer results obtained using Naïve Bayes were not completely unexpected; expected order of words to greatly impact the sarcastic nature of text; NB doesn't take ordering into account
- Based on the results of one class SVM, we can infer that using unigrams and bigrams alone for training classifiers gives large generalization error
- Both Naïve Bayes and one class SVM misclassify most of sarcastic data.
- Adding new features such as sentiment analysis, topic modelling and position tagging increased the overall prediction accuracy up to 82.2% in original SVM with Gaussian kernel.
- Probably more questions were raised than answered; further feature engineering can help produce better results

Future Work

- Re-run model leaving out one of the five feature class each time to see impact on performance using Gaussian kernel
- Apply one class SVM study to feature classes other than unigrams and bigrams
- Utilize deep learning methods

References

- Sarcasmdetector.com
- Estimating the Support of a High-Dimensional Distribution [Scholkopf et al]
- Sklearn.svm