Detecting Target of Sarcasm using Ensemble Methods

Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, David Eyers
Department of Computer Science
University of Otago
New Zealand
[pradeesh,andrew,veronica,dme]@cs.otago.ac.nz

Abstract

We describe our methods in trying to detect the target of sarcasm as part of ALTA 2019 shared task. We use combination of ensemble of classifiers and a rule-based system. Our team obtained a Dice-Sorensen Coefficient score of 0.37150, which placed 2nd in the public leaderboard. Despite no team beating the baseline score for the private dataset, we present our findings and also some of the challenges and future improvements which can be used in order to tackle the problem.

1 Introduction

We humans are complex creatures that use language as a communication tool in order to express our thoughts to one another (Sabbagh, 1999). One of the ways that we communicate with another person is through use of verbal irony. Verbal irony is defined as where the words that are being used to communicate differ from the supposed meaning (Sperber, 1984). An example of this would be from Austen (1813) Pride & Prejudice, when Darcy said to his future beloved wife, that she is "*tolerable but not handsome enough to tempt me*".

Sarcasm is a kind of verbal irony that expresses a cynical attitude towards a person or circumstance (Gibbs, 2000). In our daily lives, sarcasm is often conveyed using the tone of our voice, and/or our facial expression, to give the signal to the other person that the person is being sarcastic (Cheang and Pell, 2008). Recently, with the growth of social media many researchers have embarked on various ways of detecting sarcasm automatically. Most of their work were focused on detecting sarcasm on Twitter and on online reviews (Bamman and Smith, 2015; Rajadesingan et al., 2015; Amir et al., 2016).

Prior works treated this problem as a binary text classification problem. To the best of our knowledge, there is little work that has been done in the realm of identifying the target of sarcasm in sarcastic text. The earliest work in this domain was (Joshi et al., 2019). Identifying the target would help in certain Natural Language Processing (NLP) tasks such as in the realm of improving cyberbully detection by helping to identify the target of ridicule (Raisi and Huang, 2016). It has also sparked the organisers at the Australasian Language Technology Association (ALTA) to organise a shared challenge task to tackle the problem.

We employed a 2-phase approach to attempt to solve this task. In our *first phase*, we employed an ensemble of classifiers along with a meta-classifier to classify sarcasm targets which are marked as "*OUTSIDE*". First, we built a Support Vector Machine (SVM) using word embedding to classify the text, followed by the use of a Logistic Classifier. Finally, we used a Linear Classifier to combine the results of the two classifiers. In the *second phase* of our system, we used a rule-based approach to extract the target sarcasm words from text that are marked as "*NOT OUTSIDE*". With this proposed system, we achieved 2nd place in the public leader board of the ALTA competition. We describe our method in details in the methodology section. Next, we present our results along with some of the challenges and recommendations in improving the task. We end our paper with our plans for future work.

2 Dataset

The dataset[1] provided by the organizers of the ALTA 2019 shared task consists a of collection of sarcastic texts. There are 950 sarcastic texts for training and 544 for testing. The training dataset comes with the sarcastic text (text), along with the set of words which are the target of sarcasm (tar-

[1] https://www.kaggle.com/c/alta-2019-challenge/data

Features	Values
Number of Outsides	332
Number of Inside	618
Average Sentence Length	25.3
Average Sarcasm Target Length	3.1
Number of Subreddits	123

Table 1: Distribution and Pattern of Training Data

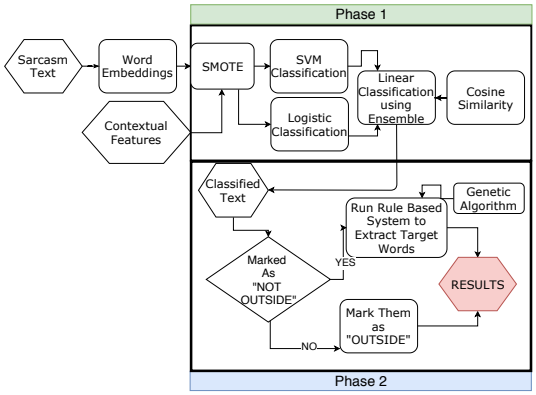

Figure 1: System Architecture

get). If the target of sarcasm is not in the text, it is marked as "*OUTSIDE*". Our task was to predict the target of the sarcasm.

We decided to analyse the training data further to understand the distribution and the pattern of the dataset. Table 1 describes the pattern. We observed that several instances of ("*NOT OUTSIDE*") have 14–19 sarcasm targets (which is half of the sentences) and other times they only have one sarcasm target. We found there to be no correlation between the sentence length and the number of sarcasm targets.

3 Methodology

We employed a 2-phase approach to tackle this problem. In the first phase, we used a series of classifiers, followed by a rule-based system in the second phase. In this section, we describe our method in detail, along with the steps that we performed. The complete system architecture is shown in Figure 1. We have also made our system's source code publicly available on GitHub.[2]

3.1 Class Imbalance

We observed that the ratio of "*OUTSIDE*" to "*NOT OUTSIDE*" in our training data set is not balanced. In order to improve our classifier's performance, we used SMOTE (Dal Pozzolo et al., 2002) to balance our dataset. SMOTE achieves this by artificially over-sampling the dataset. This has been demonstrated to improve the performance of classifiers when the dataset is small (Luengo et al., 2011).

3.2 Word Embedding

We used pre-trained model of Universal Sentence Encoding (USE) (Cer et al., 2018) to convert the text into a high-dimensional vector representation. USE is known to work well on noisy social media data. We experimented with stemming in our data to increase its accuracy, however it negatively impacted our results.

3.3 Contextual Features

We observed that our dataset was obtained from Khodak et al. (2019)'s Reddit[3] Corpus where there were both sarcastic and non-sarcastic texts present, but there was no information about the target of sarcasm. We were inspired by Wallace et al. (2014)'s work that humans require context when it comes to understanding sarcasm. In their work, when annotators were asked to classify sarcastic comments, on average 30% of the comments required annotators to ask for additional context such as the previous comment before they were able to decide. We hypothesized that we can improve our classifier's performance by adding additional context extracted from Khodak et al. (2019)'s corpus to our original dataset.

We converted each Subreddit label found in Khodak et al. (2019)'s dataset into categorical data values using one-hot encoding. For categories that were not present in both training and testing data, we grouped them together into a category known as "Others". We have also extracted the number of likes and dislikes on each post. They are continuous features, we used Z-Score normalization to improve our classifier's performance (Jayalakshmi and Santhakumaran, 2011)

$$x' = \frac{x_1 - \mu_1}{\sigma_1} \quad (1)$$

[2] https://github.com/prasys/sarcasm-detection

[3] Reddit http://www.reddit.com is a social news aggregation and discussion website

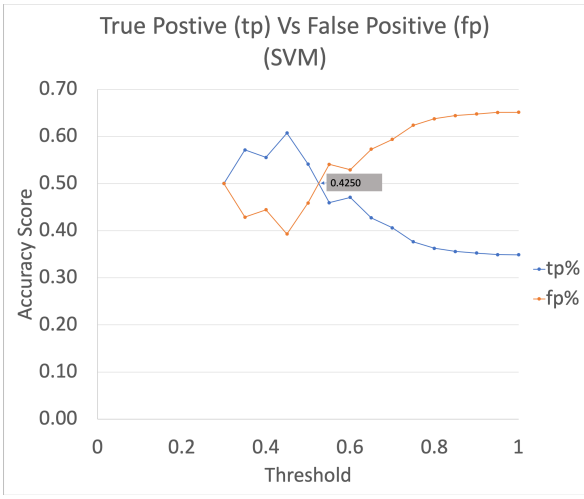

Figure 2: Recall Score against Threshold for SVM Classifier

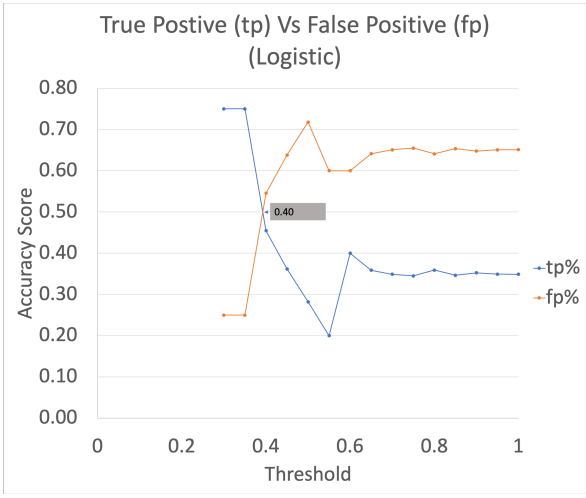

Figure 3: Recall Score against Threshold for Logistic Classifier

where x_1 is the value of the feature, μ_1 is the mean value of the feature in training data and σ_1 is the standard deviation for the feature in the training data.

3.4 Phase 1

In our first phase, we used a Support Vector Machine (SVM) with word embedding from USE as its input for our SVM Classifier. SVM has been known to perform very well on high dimensional input vectors (Goudjil et al., 2018). We experimented with other classifiers such as Logistic Regression and Random Forest but it did not yield good results. For our SVM Classifier, we set the classification's threshold value to be 0.425 and above in order for the text to be classified as "*OUTSIDE*". This was done to minimize the false positives. Figure 2 shows the various thresholds and the accuracy score regarding true positive (TP) and false positives (FP).

The additional data features that we have extracted from Khodak et al. (2019)'s corpus are used as input vectors for our logistic classifier. Just like our SVM Classifier, we fine-tuned our logistic classifier's threshold value to be 0.40 and above for a text to be classified as "*OUTSIDE*". Figure 3 shows the performance of the classifier. The values for both of the classifiers were obtained by performing 3-fold cross-validation.

We introduced cosine similarity to further strengthen the meta-classifier's performance. It is calculated by using the word embedding we obtained earlier. If we obtain a similarity score of 0.70 or higher, we assign a score of 1 otherwise a score of 0.

Rules	Rule No
R1	Pronouns & pronominal adjectives
R2	Named entities
R3	Object of a positive sentiment verb
R4	Phrase on negative side of verb
R5	Gerund & infinitive verb phrases
R6	Nouns after positive sentiment adjective
R7	Subject of interrogative sentences
R8	Subjects of comparisons (similes)
R9	Demonstrative adjective-noun pairs

Table 2: Definition of the Rules for the Rule-based Component within the Proposed System

Finally for our meta classifier, we used a Linear Classification (Džeroski and Ženko, 2004).

We used the probability scores from both of the classifiers and cosine similarity as input vectors into the classifier. We did not fine-tune the linear classifier and used the default value of 0.5 and above to classify text as "*OUTSIDE*".

3.5 Phase 2

In our second phase, we used the rule-based system to extract the target of sarcasm from the texts. The rules that we used are described in Table 2, and adopted from (Joshi et al., 2019). We implemented the rules using NLTK Toolkit.[4]

We applied some minor adjustments to R1 and R2 that increased the performance 4.49% and 39.68% respectively, over the original rules, as described below.

For R1, we included the subject of each pronoun. For example in the training set one of the

[4] https://www.nltk.org/

Rules	DSC Score
R1 (Without Subject)	0.2696
R1 (With Subject)	0.2817
R2 (Without Truecase)	0.0814
R2 (With Truecase)	0.1137
R3	0.0266
R4	0.0800
R5	0.1094
R6	0.0598
R7	0.0766
R8	0.0105
R9	0.0196

Table 3: Performance of Rules Score

target of sarcasm was identified as "*you,op*".[5] The original rule set would only identify "*you*".

As for R2, in order to get the Named Entities (NE) recognized, we used Truecasing (Lita et al., 2003). This helped to correct the case of our noisy data which further improved NE recognition. Lowering all the cases does not work as it presents a problem in distinguishing named entities from nouns. For example, the word "*apple*" may be interpreted as the fruit and not the company. However due to time constrains, we did not take a look at other rules in-depth but intend to do so as future work.

In order to determine how effective each rule was, we ran the rules one by one over the training data after excluding all the text which were marked as "*OUTSIDE*". We used Dice-Sorensen Coefficient (DSC) in order to measure the performance.

$$D(A, B) = 2 \times \frac{A \cap B}{|A| + |B|} \quad (2)$$

where A are predicted words and B are actual words.

Table 3 shows the individual performance for each rule. In order to determine which rules were likely to give us the high scores, we implemented a genetic algorithm to obtain weights for each of the rules. We ran our genetic algorithm across 500 generations with 80% probability of mutation. Figure 4 shows the performance of our genetic algorithm. The algorithm assigned a good weighting scores for R1, R2, R3, and R5 respectively. For the other rules, negative weighting scores were given.

4 Results

We investigated the results and the behavior of the system by submitting our runs to the competition.

[5]OP is an abbreviation for Original Poster

System	Public Score	Private Score
Baseline (OUTSIDE)	0.36764	**0.34926**
Baseline (Pronoun)	0.20933	0.22539
SVM (Stemming) + Rules	0.30203	0.26553
SVM + Rules	0.35983	0.30777
Logistic + Rules	0.11397	0.12867
Ensemble + Rules	0.36889	0.30027
Ensemble + Tuned Rules	**0.37150**	0.29134

Table 4: System Evaluation

Kaggle is used as the platform for submission of runs. In Kaggle, the training data provided to us by ALTA organizers is split into public (public leaderboard) and private (private leaderboard). The private portion serves as a validation portion in order for the organizers to determine the effectiveness of the system. The scores are evaluated by using DSC Score (Equation 2). We summarise and present our results in Table 4.

4.1 Discussion

The objective set by the organizers at ALTA was to beat the two baselines provided by them. The first baseline always predicted "*OUTSIDE*". The second one always predicted the pronouns from the text as the target for the "*NOT OUTSIDE*" text. Our system beat both baselines for the public leaderboard, but we did not manage to beat the baseline for private score. In fact, no teams beat the scores in the private baseline. Prior to proposing our final system, we have built and evaluated various different systems which included just using one classifier which is either SVM or Logistic Regression and the rule-based system. Then we used the ensemble of classifiers. We believe that our ensemble classifiers performed poorly on the

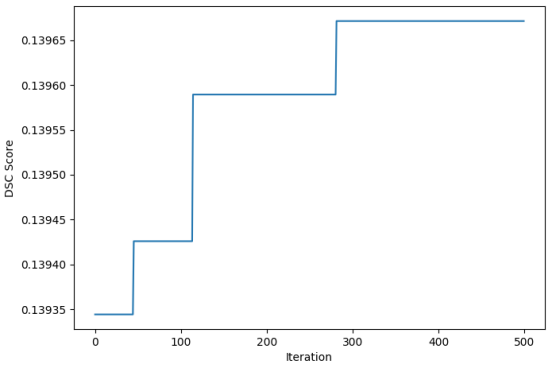

Figure 4: Performance of genetic algorithm across 500 generations

Predicted Words	DSC Score
we die out	0.5454
Entire Sentence	0.4705
sun gonna destroy us we die	0.2828
sun die	0.2000

Table 5: DSC Scores

private score as it might have been biased to the public data. On the other hand, just using the additional features alone to classify yielded poor results as our system could not identify "*OUTSIDE*" accurately.

This prompted us to look deeper into the problem and offering several ways on how it can be addressed. We discuss this in subsection 4.2 and 4.3.

4.2 Evaluation Metric

Based on equation 2, we can deduce that the score for predicting "*OUTSIDE*" would be easier to obtain compared to predicting the target of sarcasm words correctly which may be trickier. In order to demonstrate our point, let us look at the following two examples which we took from training data. The target of sarcasm given by the judges are highlighted in bold

> "*Oh man and while we are at it we can make it so when the boss dies you can hand pick the piece of gear you want!*." ("*OUTSIDE*")

> "*The sun is gonna destroy us in a few billion years anyways, so why does it matter if* ***we die out in the next few centuries****?*."

In the first example the target of sarcasm is outside. DSC score would yield a perfect score of 1 if it predicted properly. In the second example, it is very hard to get a perfect DSC of 1. In Table 5, we show how the score varies depending on the number of words predicted correctly, and length of the predicted words. We can clearly see that it is very challenging to get a very high score even when we can predict all of the relevant targets.

One way of addressing the performance of the system is to use accuracy score as an additional metric to determine the effectiveness of the system. This would also help to gauge the capacity of the systems identifying true positives (TP) and true negatives (TN).

4.3 Human Perspective & Relevance Judgement

In their works, (Joshi et al., 2016) have highlighted some of the difficulties that annotators face in identifying sarcasm and irony. From our failure-analysis, we have determined that humans' annotations can be inconsistent. We show two of the examples from the training dataset, with the target of sarcasm annotated by the judges in bold.

> "***OP*** *is just some white knight who always comes to the aid of the female, if you knew her you'd know how much of a whore she is.*."

> "*$10* ***OP*** *wants to do something crazy with trading cards and is just trying to get you all to sell them to* ***him*** *on the cheap*"

In the first example, we can clearly make the association that "*you*" from the first example refers to "*OP*" but only "*OP*" is identified as the target of sarcasm. However, in the second example, both the words "*OP,him*" are identified as the target of sarcasm by the judges. This shows to us that even in sentences which are constructed in a similar manner, the way judges identify the target of sarcasm differs from one person to another.

In order to address this gap, we propose that additional assessments should be conducted. For example, in the Text Retrieval Conference (TREC), participants submit their assessments and let the human annotators decide if the documents retrieved by the search engines were relevant to the given queries (Hawking et al., 1999). We believe that adopting this approach for this task instead of the current approach would help to address the shortcomings of relying entirely on human annotators.

5 Conclusion and Future Work

We presented an approach to identify the target of sarcasm. We competed in the ALTA 2019 Competition under the team name of "*orangutan*". Our best-performing system used an ensemble of classifiers. Despite achieving a score of 0.37150 and beating the baselines in the public portion within Kaggle, we did not manage to beat the baseline in the private dataset.

We believe that there is still much work to be done in this domain. As part of future work we are

planning to tackle this problem in several ways, including:

- Improving our classifier;
- Further improving the rule-based system; and
- Experimenting with deep learning models.

Acknowledgments

Many thanks to Kat Lilly, Dr.Diego Moll-Aliod for their time in proofreading this paper. We would also like to thank the ALTA organizers for their support.

References

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mario J. Silva. 2016. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 167–177.

Jane Austen. 1813. *Pride and Prejudice*. Routledge/Thoemmes, London.

David Bamman and Noah A. Smith. 2015. Contextualized sarcasm detection on twitter. *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pages 574–577.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *Empirical Methods in Natural Language Processing*.

Henry S. Cheang and Marc D. Pell. 2008. The sound of sarcasm. *Speech Communication*, 50(5):366–381.

A. Dal Pozzolo, O. Caelen, and G. Bontempi. 2002. Comparison of balancing techniques for unbalanced datasets. *Journal of Artificial Intelligence Research 16*, 16(1):732–735.

Saso Džeroski and Bernard Ženko. 2004. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273.

Raymond W Gibbs. 2000. Metaphor and Symbol Irony in Talk Among Friends Irony in Talk Among Friends. *Metaphor and Symbol*, 15(2):1–2.

Mohamed Goudjil, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. A Novel Active Learning Method Using SVM for Text Classification. *International Journal of Automation and Computing*, 15(3):290–298.

David Hawking, Nick Craswell, and Paul Thistlewaite. 1999. Overview of TREC-7 Very Large Collection Track. In *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7)*, pages 1–13.

T. Jayalakshmi and A. Santhakumaran. 2011. Statistical Normalization and Back Propagationfor Classification. *International Journal of Computer Theory and Engineering*, 3(1):89–93.

Aditya Joshi, Pranav Goel, Pushpak Bhattacharyya, and Mark J. Carman. 2019. Sarcasm target identification: Dataset and an introductory approach. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, (2008):2676–2683.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, Mark Carman, Meghna Singh, Jaya Saraswati, and Rajita Shukla. 2016. How Challenging is Sarcasm versus Irony Classification?: A Study With a Dataset from {E}nglish Literature. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 123–127.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2019. A large self-annotated corpus for sarcasm. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 641–646.

Lucian Vlad Lita, I B M T J Watson, and I B M T J Watson. 2003. tRuEcasIng. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 152–159.

Julián Luengo, Alberto Fernández, Salvador García, and Francisco Herrera. 2011. Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing*, 15(10):1909–1936.

Elaheh Raisi and Bert Huang. 2016. Cyberbullying Identification Using Participant-Vocabulary Consistency. In *ICML Workshop on #Data4Good: Machine Learning in Social Good Applications*, pages 46–50, New York.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter. In *WSDM '15: Proceedings of the Eight ACM International Conference on Web Search and Data Mining*, pages 97–106.

Mark A. Sabbagh. 1999. Communicative intentions and language: Evidence from right-hemisphere damage and autism. *Brain and Language*, 70(1):29–69.

Dan Sperber. 1984. Verbal irony: Pretense or echoic mention? *Journal of Experimental Psychology: General*, 113(1):130–136.

Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 2, pages 512–516.