ISD at SemEval-2022 Task 6: Sarcasm Detection Using Lightweight Models

Samantha Huang
Lynbrook High School
Saratoga, CA
sam.y.huang2005@gmail.com

Ethan A. Chi
Stanford University
Stanford, CA
ethanchi@cs.stanford.edu

Nathan A. Chi
De Anza College
Cupertino, CA
chinathan
@student.deanza.edu

Abstract

Robust sarcasm detection is critical for creating artificial systems that can effectively perform sentiment analysis in written text. In this work, we investigate AI approaches to identifying whether a text is sarcastic or not as part of SemEval-2022 Task 6. We focus on creating systems for Task A, where we experiment with lightweight statistical classification approaches trained on both GloVe features and manually-selected features. Additionally, we investigate fine-tuning the transformer model BERT. Our final system for Task A is an Extreme Gradient Boosting Classifier (XGB Classifier) trained on manually-engineered features. Our final system achieved an F1-score of 0.2403 on Subtask A and was ranked 32 of 43.

1 Introduction

Sarcasm is the use of irony–which communicates the opposite of what is said–to humorous and derisive effect (Bouazizi and Ohtsuki, 2016). On the web, sarcasm is ubiquitous–not least because social media users often apply sarcasm to incorporate a sardonic sense into their statements (Hancock, 2004). This poses a substantial challenge to artificial systems evaluating tasks including sentiment analysis (Liu and Zhang, 2012). It is already a challenge enough for human annotators to determine what is intended to be taken at face-value or not in context-lacking text; it is even more difficult for NLP systems to distinguish between what should be taken literally and what is sarcastic.

Task 6 of SemEval-2022 (Abu Farha et al., 2022) provides an environment to build systems to approach these challenges. In particular, Subtask A in Task 6 (Table 1) of SemEval-2022 tests the ability of automated systems to determine whether a text is sarcastic or non-sarcastic. We investigate whether lightweight models (which use few computational resources) are able to effectively identify sarcastic speech; we also experiment with fine-tuned Transformer-based models to identify whether larger models perform better.

2 Dataset

In Subtask A, we train our models on the official SemEval-2022 Task 6 English training set, which was curated from a set of tweets. Each sentence (examples in Table 6) has been annotated for sarcasm status by the text authors themselves, with 1 denoting a sarcastic text and 0, a non-sarcastic text.

Task	Description	Metric
A	Determine whether a given text is sarcastic or non-sarcastic.	F1, sarcastic-class

Table 1: Subtask A overview.

2.1 Train-test Split

The dataset has a total of 4868 examples, with 3468 being part of the training set, and 1400 being part of the test set. In total, there are 867 sarcastic and 2601 non-sarcastic texts in the training set. As the testing set labels were not provided until after the competition, we created our own validation set with a 75 : 25 train : test split. Thus, our train set has 2774 examples and our test set has 693 examples.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 919 - 922

Feature	Description	LR Coefficient
POL	Words referring to political leaders (e.g. "Boris," "Trump").	0.127
GAY	The word "gay."	0.008
BANG	The character "!".	-0.067
AT	The character "@".	0.005
DEFINITELY	The word "definitely."	0.035
PLEADING FACE	The pleading face emoji.	0.136
EMOJI	The grinning face emoji.	0.095
HASH	The character "#".	-0.095
THANK	The word "thank."	0.008
HAHA	The word "haha."	0.095

Table 2: Manual features for sarcasm detection.

Model	F1 Sarcastic	F-score	Precision	Recall	Accuracy
XGBClassifier	0.2403	0.1332	0.3651	0.4792	0.1464

Table 3: Official test set performance of our best-performing lightweight model (XGBClassifier trained with manual features) on Subtask A (binary classification). LR coefficient represents the linear regression coefficient value for the given feature.

3 Methods

3.1 Subtask A: Sarcasm Detection

This subtask examines whether a given text is sarcastic, and we investigate using the following models. Our lightweight machine learning models were implemented using the Scikit-learn library (Pedregosa et al., 2011):

- **Logistic Regression** is a supervised learning algorithm that predicts a binary outcome using a logistic function.
- **GaussianNB** is a type of Naive Bayes algorithm used for continuous data that follows a normal distribution (Qiu et al., 2020).
- **SVM** is a non-probabilistic binary linear supervised learning algorithm that can be used for classification and regression (Yu and Kim, 2012).
- **AdaBoostClassifier** is a meta-algorithm that assigns higher weights to incorrectly classified samples to improve the following classifiers (Solomatine and Shrestha, 2004).
- **XGBClassifier** stands for e**X**treme **G**radient **B**oosting **Classifier** and is a decision tree based algorithm that uses gradient boosting methods to avoid overfitting (Kumar et al., 2021).
- **BERT** is a transformer based algorithm that uses masked language modeling. We fine-tune BERT–an approach commonly used in tasks such as sentiment prediction–which was pretrained on language modelling and next-sentence prediction tasks. In particular, we use BERT base cased, BERT large cased, BERT base uncased, and BERT large uncased (Devlin et al., 2018).

3.2 Results

On the unofficial evaluation set, XGBClassifier performed the best compared to other models. On the official evaluation set, we achieve a F1-score of 0.2403. We were ranked 32 out of 43. Our official and unofficial results are listed in Table 3 and Table 4 respectively. The hyperparameters that we used for all models trained on manual features is included in Table 5.

4 Conclusion

Our models were trained to determine whether texts were sarcastic or not. For the most part, our models struggled to detect sarcasm in text—-as was expected, given that the task was quite challenging even for humans. We find that the models that achieve the highest degree of success in detecting sarcasm were GaussianNB and XGBClassifier models.

Model	Features	positive-class F1	Accuracy	Normalize
LogisticRegression	Manual	0.44	0.34	True
LogisticRegression	GloVe	0.09	0.71	False
GaussianNB	Manual	0.43	0.34	True
GaussianNB	GloVe	0.42	0.47	False
SVM	Manual	0.30	0.63	True
SVM	GloVe	0.08	0.72	False
AdaBoostClassifier	Manual	0.06	0.72	False
AdaBoostClassifier	GloVe	0.23	0.68	False
XGBClassifer	**Manual**	**0.45**	**0.32**	**False**
XGBClassifier	Glove	0.38	0.59	False
BERT base cased	–	0.32	0.63	False
BERT large cased	–	0.14	0.50	False
BERT base uncased	–	**0.40**	**0.75**	**False**
BERT large uncased	–	0.26	0.63	False

Table 4: Unofficial validation set performances of candidate models. For this task, the highest-performing lightweight model is XGBClassifier and the highest-performing transformer model is BERT base uncased.

We also find that using manual features, as listed in Table 2, is a fruitful approach to determining the sarcasm status of a sentence. In particular, we preprocess the data by identifying the number of instances of characters or words described in each feature category, then train our models on these summed feature values. Our top-scoring classifiers yielded substantially greater postive-class F1 scores with manual features than with automatic GloVe features. That being said, it should be noted that using these manual features also lowered the accuracy greatly, which indicates a tradeoff between F1 score and accuracy due to the extreme class imbalance of the dataset.

Finally, fine-tuning BERT achieves reasonable results while detecting sarcasm. However, this method is still inferior to a lightweight approach.

Overall, our best model, the XGBClassifier with manually engineered features, did not perform significantly better than the Logistic Regression model. Our results demonstrate that boosting algorithms can predict sarcasm in text to a moderate degree of success.

Acknowledgments

The authors would like to acknowledge Google Colaboratory for their free compute services.

References

Ibrahim Abu Farha, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Mondher Bouazizi and Tomoaki Otsuki Ohtsuki. 2016. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:5477–5488.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeffrey T Hancock. 2004. Verbal irony use in face-to-face and computer-mediated conversations. *Journal of Language and Social Psychology*, 23(4):447–463.

Munish Kumar, Manish Kumar, et al. 2021. Xgboost: 2d-object recognition using shape descriptors and extreme gradient boosting classifier. In *Computational Methods and Data Engineering*, pages 207–222. Springer.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Ming Qiu, Yiru Zhang, Tianqi Ma, Qingfeng Wu, and Fanzhu Jin. 2020. Convolutional-neural-network-based multilabel text classification for automatic discrimination of legal documents. *Sens. Mater*, 32(8):2659–2672.

Dimitri P Solomatine and Durga L Shrestha. 2004. Adaboost. rt: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 1163–1168. IEEE.

Hwanjo Yu and Sungchul Kim. 2012. Svm tutorial-classification, regression and ranking. *Handbook of Natural computing*, 1:479–506.

Model	Hyperparameter	Task 1a
GaussianNB	priors	0.025, 0.975
	var_smoothing	1e-09
SVM	class_weight	balanced
	C	1.0
	kernel	rbf
	degree	50
AdaBoostClassifier	base_estimator	max_depth = 1 class_weight = {0: 0.1, 1: 0.9}
	$n_{estimators}$	50
	learning_rate	1.0
	loss	linear
XGBClassifier	$n_{estimators}$	100
	max_depth	5
	eta	0.3
	min_child_weight	1
	booster	gbtree

Table 5: Hyperparameters for best-performing Manual models.

Sentence	Sarcastic
yeah your girl is fine but does she pass out while giving blood	1
just impulse bought a mandolin and in 3-5 buisness days i will impulse learn some jigs	0

Table 6: Examples that are sarcastic and not sarcastic, respectively.