UTNLP at SemEval-2022 Task 6: A Comparative Analysis of Sarcasm Detection using generative-based and mutation-based data augmentation

Amirhossein Abaskohi, Arash Rasouli, Tanin Zeraati, Behnam Bahrak
School of Electrical and Computer Engineering, College of Engineering
University of Tehran, Tehran, Iran
{amir.abaskohi, arash.rasouli, t.zeraati, bahrak}@ut.ac.ir

Abstract

Sarcasm is a term that refers to the use of words to mock, irritate, or amuse someone. It is commonly used on social media. Because sarcasm has a negative influence on society harmony and peace, a strategy to stop sarcasm from spreading online is needed. The technique and results of our team, UTNLP, in the SemEval-2022 shared task 6 on sarcasm detection are presented in this paper. We put different models and data augmentation approaches to the test and report on which one works best. The tests begin with fundamental machine learning models and progress to transformer-based and attention-based models. We employed data augmentation based on mutations and generative data. Our best approach, using RoBERTa and mutation-based data augmentation, acheived a F1 score of 0.38 in the competetion's evaluation phase.

1 Introduction

Billions of internet users use social networks not only to stay in touch with friends, meet new people, and share user-generated content, but also to express and share their opinions on a wide range of topics using a variety of methods such as posting comments, videos, photos, and other information with specific groups of people(Tungthamthiti et al., 2016). In these platforms, users were able to submit information on whatever topic they wanted, with no restrictions on the sort of content they may share. The lack of constraints, along with individuals' anonymity on these networks, led in humorous data.

Because sarcasm indicates sentiment, detecting sarcasm in a text is critical for anticipating the text's accurate sentiment, making sarcasm detection a valuable tool with multiple applications in domains such as security, health, services, product evaluations, and sales. Sarcasm detection is an important aspect of creative language comprehension(Veale et al., 2019) and online opinion mining (Kannangara, 2018). Even for humans, identifying sarcasm is difficult due to heavily contextualized expressions(Walker et al., 2012). There are few labeled data resources for sarcasm detection, and any available texts that can be collected (for example, Tweets) contain many issues, such as an evolving dictionary of slang words and abbreviations, necessitating many hours of human annotation to prepare the data for any potential use. Furthermore, the nature of sarcasm identification adds to the difficulty of the task, as sarcasm may be considered relative and varies greatly across persons, depending on a variety of criteria such as the topic, area, time, and events surrounding the statement.

In an attempt to solve this issue, we participated in SemEval-2022 shared task 6(Abu Farha et al., 2022), which aims to recognize whether a tweet is sarcastic or not. The following are our contributions: We start by experimenting with simple machine learning models like Support Vector Machines (SVM) and various word encodings. Then, in order to discover the optimum data preparation method, we test the effect of data preprocessing. On our best data set, we evaluate Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) based models such as Long Short Term Memory (LSTM) and Deep Bidirectional LSTM (DBLSTM). Following that, we put deep Bidirectional Encoder Representations from Transformers(BERT)-based and attention-based models to the test. Finally, we put several data augmentation techniques to the test. Different neural network topologies and embeddings are compared, and the model with the highest performance is reported. With RoBERTa(A Robustly Optimized BERT Pretraining Approach) and mutation-based data augmentation, our top result gets an F1 of 0.38. However we obtain better outcomes, with a 0.414 F1 score.

2 Previous Work

Sarcasm detection has been represented as a binary classification issue, with the majority of tweets labeled with certain hashtags (e.g., sarcasm, sarcastic) being considered sarcastic. Many techniques in various languages have been proposed using this framework.

In (Davidov et al., 2010), semi-supervised sarcasm detection experiments were done using a Twitter dataset (5.9 million tweets) and 66,000 Amazon product evaluations. On the product reviews dataset, they acquired an F-measure of 0.83 and on the Twitter dataset, they obtained an F-measure of 0.55 using 5-fold cross validation on their kNN-like classifier. They utilized the hashtag sarcasm as an indication of sarcastic messages while obtaining the Twitter dataset.

(González-Ibánez et al., 2011) used 900 messages from Twitter sorted into three groups (sarcastic, positive sentiment, and negative sentiment). To find sarcastic tweets, they utilized the hashtags sarcasm and sarcastic. Support Vector Machine (SVM) with Sequential Minimum Optimization (SMO) and logistic regression were employed as classifiers. The best result best accuracy for sarcastic class was 0.65.

(Reyes et al., 2012) presented elements to capture ambiguity, polarity, unexpectedness, and emotive situations in figurative language. F-measure 0.65 was the best result in the categorization of irony and general tweets.

The representativeness and significance of conceptual elements have been investigated in (Reyes et al., 2013). Punctuation marks, emoticons, quotations, capitalized words, lexicon-based features, character n-grams, skip-grams, and polarity skip-grams are all examples of these characteristics. Each of the four categories (irony, comedy, education, and politics) in their corpus has 10,000 tweets. Using the Naive Bayes and decision trees algorithms, they evaluated two distributional scenarios: balanced distribution and unbalanced distribution (25 percent ironic tweets and 75 percent tweets from the three non-ironic categories). The decision trees classified the balanced distribution with an F-measure of 0.72 and the unbalanced distribution with an F-measure of 0.53.

One sort of sarcasm identified by (Riloff et al., 2013) is the difference between a good mood and a bad scenario. They gathered collections of positive sentiment phrases and negative circumstance words from sarcastic tweets using a bootstrapping approach. They suggested a method for classifying tweets as sarcastic if they contain a positive predicative in close proximity to a negative context phrase. They used the SVM classifier using unigrams and bigrams as features to evaluate a human-annotated dataset3 of 3000 tweets (23 percent sarcastic), getting an F-measure of 0.48. The F-measure of the hybrid strategy, which combined the findings of the SVM classifier with their contrast method, was 0.51.

(Lukin and Walker, 2017) used bootstrapping, syntactic patterns, and a high precision classifier to classify sarcasm and nastiness in online chats. On their snark dataset, they got an F-measure of 0.57.

In (Oprea and Magdy, 2019), LSTM, Att-LSTM, CNN, SIARN, MIARN, 3CNN, and Dense-LSTM models were used to assess the task dataset that was introduced in citeoprea2019isarcasm, which is an unbalanced dataset and labeled by the tweets' writers. Using the MIARN model, they were able to get an F-score of 0.364.

In (Guo et al., 2021), the Latent Optimized Adversarial Neural Transfer (LOANT) model was suggested as a novel latent-optimized adversarial neural transfer model for crossdomain sarcasm detection. LOANT surpasses classical adversarial neural transfer, multitask learning, and meta-learning baselines by using stochastic gradient descent (SGD) with onestep look-ahead, and sets a new state-of-the-art F-score of 0.4101 on the iSarcasm dataset.

3 Data

We mostly used the Isarcasm (Oprea and Magdy, 2019) dataset in this study. In certain experiments, we integrated the primary dataset with various secondary datasets, including the Sarcasm Headlines Dataset (Misra and Arora, 2019) and Sentiment140 dataset (Go et al., 2009) to increase the quantity of data and compensate for the lack of sarcastic data. For each dataset, the details are further discussed.

3.1 Main Task Dataset: Isarcasm

According to (Oprea and Magdy, 2019), the sarcasm labeling using hashtag to building datasets captures just the sarcasm that the annotators were able to detect, leaving out the intended sarcasm. When the author intends for the content to be sarcastic, it is called intended sarcasm. The dataset includes 4484 tweets, 3707 non-sarcastic and 777 sarcastic. Because some tweets had been erased, we only had access to 3469 tweets for the job. The unbalanced dataset and the scarcity of sarcastic data were two of the most significant issues we encountered. Table 1 displays some of the dataset's annotated remarks.

3.2 Sarcasm Headlines Dataset

Sarcasm Headlines Dataset(Misra and Arora, 2019) was gathered from two news websites. It is beneficial since it overcomes the constraints of Twitter datasets due to noise. As the second edition of this dataset includes more data and a greater variety of data than the first version, we chose the second version.

3.3 Sentiment 140 Dataset

We needed to compensate for the limited data to train our model successfully. As a result, we chose the sentiment 140 dataset (Go et al., 2009) because of it has large quantity of data and is on Twitter. The sentiment tweet message is labeled using an automated classification approach in this dataset. The accuracy is more than 80% when using a machine learning algorithm.

4 Methodology

We examined and analyzed various models and data augmentation strategies for sarcasm detection in this study. First, we will go through data augmentation methods, then We will go through the structure and hyperparameters of these models in this section. The codes of all models are freely available on GitHub[1].

4.1 Data Augmentation

4.1.1 Generator-based

For this augmentation method, we used GPT-2(Radford et al., 2019) generative model to generate 4000 tweets for both sarcastic and non-sarcastic classes. Then we selected 2000 tweets of each class uniformly at random to increase dataset quantity and have more sarcastic samples.

4.1.2 Mutation-based

In this method, we used three distinct ways to change the data: eliminating, replacing with synonyms, and shuffling. These processes were used in the following order: shuffling, deleting, and replacing. The removal and replacement were carried out in a systematic manner. We used the words' roots to replace synonyms and created a lexicon based on the roots. When a term was chosen to be swapped with its synonyms, we chose one of the synonyms at random(Figure 1). We tried each combination of these processes to find the best data augmentation combination (a total of seven).

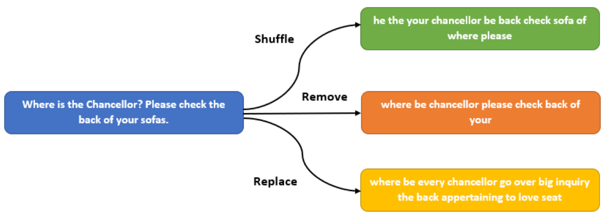

Figure 1: Effect of shuffling, word elimination, and replacing with synonyms on a tweet sample.

4.2 Models

4.2.1 Support Vector Machine (SVM)

We utilized SVM to discover the optimal approaches for dataset preparation and word embeddings in this section. For data augmentation, we employed both generator-based and mutation-based methods. We also put other data preparation approaches to the test, such as link removal, emoji removal, stop word removal, stemming, and lemmatizing. We utilized TF-IDF, Word2Vec(Mikolov et al., 2013), and BERT(Devlin et al., 2018) for word embedding.

The findings revealed that using a regularization value of 10 and a Radial Basis Function (RBF) kernel, BERT word embedding and no data preprocessing will give us the best results.

4.2.2 LSTM-based Methods

We begin with the intuition that a model with memory can help us to reach better results. So we started with Long Short Term Memory(LSTM) model(Hochreiter and Schmidhuber, 1997). We used one LSTM layer followed by time distributed dense layer. We repeated these two layers one more time and then we used another LSTM layer followed by two dense layers. This model and all of the following models in this section, were trained in 10 epochs.

In addition, we used Bidirectional Long Short Term Memory(BLSTM) too. Using bidirectional will run your inputs in two directions, one from past to future and the other from future to past. For this network we used one bidirectional LSTM

[1] https://github.com/AmirAbaskohi/SemEval2021-Task6-Sarcasm-detection

Table 1: Example of Sarcastic and Non-Sarcastic tweets.

Tweet	Sarcastic	Sarcasm Type
Oh my goodness. It's the first week of the summer holidays and Harrison has found his recorder Give.Me.Strength.	Sarcastic	['Sarcasm']
90% of adulthood is just refilling your Brita pitcher.	Sarcastic	['Irony', 'overstatement']
True bliss is laying in an ice cold bath during the hottest part of the year	Non-Sarcastic	[]

layer followed by time distributed dense layer. We repeated these two layers one more time and then we used another bidirectional LSTM layer followed by two dense layers.

Furthermore, we combined LSTM and BLSTM with Convolutional Neural Networks (LSTM). Convolutional Neural Network (CNN) layers for feature extraction on input data are paired with LSTM to facilitate sequence prediction in the CNN-LSTM architecture. Despite the fact that this model is often employed for video datasets, (Rehman et al., 2019) demonstrated that it can perform better in sentiment analysis tasks. We used three 1D concolutional layers followed by 1D global max pooling layer for the convolutional part. We used these layers at the end of LSTM-based networks.

4.2.3 BERT-based Methods

The use of the bidirectional training of Transformer, a prominent attention model, to language modeling is BERT's fundamental technological breakthrough(Devlin et al., 2018). The findings of the research suggest that bidirectionally trained language models can have a better understanding of language context and flow than single-direction language models. The researchers describe a new approach called Masked LM(MLM) that permits bidirectional training in models that were previously difficult to train in.

Robustly Optimized BERT or RoBERTa has a nearly identical architecture to BERT, however the authors made some minor adjustments to its architecture and training technique to enhance the results on BERT architecture(Liu et al., 2019).

We used both RoBERTa with twitter-roberta-base, which is on near 58 million tweets and fine-tuned for sentiment analysis with the TweetEval benchmark, and BERT with bert-base from Huggingface(Wolf et al., 2019). For both models, we employed 5 epochs, batch size of 32, 500 warmup steps, and weight decay of 0.01.

4.2.4 Attention-based Methods

One of the most important achievements in Deep Learning research in the previous decade is the attention mechanism(Vaswani et al., 2017). The Encoder-Decoder model's restriction of encoding the input sequence to one fixed-length vector from which to decode each output time step is addressed via an attention mechanism. This difficulty is thought to be more prevalent when decoding extended sequences.

We start with the assumption that if a model with an attention layer is trained to identify sarcasm at the sentence level, the sarcastic words will be the ones the attention layer learns to value. As a result we added attention layer to our LSTM-based and BERT-based models. The results will be discussed further.

4.2.5 Google's T5

Google's T5(Raffel et al., 2019) text-to-text model outperformed the human baseline on the GLUE, SQuAD, and CNN/Daily Mail datasets, and earned a remarkable 88.9 on the SuperGLUE language benchmark.

We fine-tuned T5 on our problem and our dataset with giving the sarcastic label as target, and giving the tweets as source. We used 2 epochs, batch size of 4, 512 tokenization max length, Adam epsilon of 1e-8, word decay of 0, 0 warmup steps and learning rate of 3e-4[2](Figure 2).

5 Results

In this section we will report the results of our models introduced in previous section.

[2]We were not able to test a larger version of the model due to system constrain

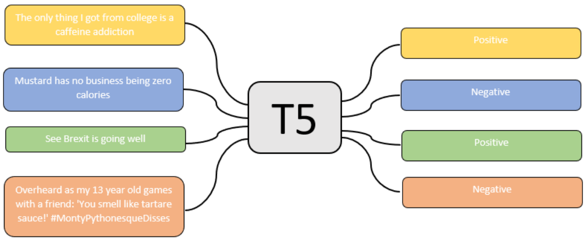

Figure 2: Fine-tuning T5 model for sarcasm detection problem.

Table 2: F1-score and accuracy for different data augmentation methods on SVM model with BERT word embedding and no preprocessing.

Data Augmentation	F1-Score	Accuracy
Shuffling	0.305	0.7471
Shuffling + Replacing	0.3011	0.7414
Shuffling + Elimination	0.3064	0.7478
Elimination	0.301	0.7478
GPT-2	0.2923	0.675

5.1 Support Vector Machine (SVM)

The optimum augmentation technique, preprocessing method, and word embedding were all determined using the SVM model. Without any augmentation, BERT obtained the greatest F1-score of 0.2862, compared to 0.2541 and 0.0924 for Word2Vec and TF-IDF, respectively.

We have also looked at several ways of data augmentation. The F1-scores for shuffling with replacing words, only word elimination, just shuffling, and shuffling with word elimination were the highest in the mutation-based augmentation. We tried these data augmentation and GPT-2 data augmentation on RoBERTa as well because the results were so close, and we discovered that merely word removal was the best data augmentation. All of the results below are based on no data preparation, BERT word embedding, and mutation-based data augmentation utilizing just word removal.

5.2 LSTM-based Methods

Because our models for this portion were not very intricate, our results aren't particularly impressive. LSTM obtained F1-score 0.2176 using BERT word embeddings, mutation-based data augmentation, and no preprocessing, whereas BLSTM achieved F1-score 0.2439 using BERT word embeddings, mutation-based data augmentation, and no preprocessing. The F1-score of the LSTM was increased to 0.2453 and the BLSTM was increased to 0.2751. The CNN model's F1-score was 0.2263.

Table 3: Best results for each model using iSarcasm dataset and mutation-based data augmentation.

Model	F1-Score	Accuracy
SVM	0.3064	0.7478
LSTM-based	0.2751	0.7251
BERT-based	0.414	0.8634
Attention-based	0.2959	0.7793
Google's T5	0.4038	0.8124

5.3 BERT-based Methods

We employed a mutation-based data augmentation approach with no preparation for BERT-based procedures. We got an F1-score of 0.323 using BERT. We achieved our best result with RoBERTa, which was an F1-score of 0.414, which was better than LOANT(Guo et al., 2021) model.

We also evaluated the Sarcasm Headlines and Sentimend140 datasets, however the F1-score was lower, which we assume was due to the differences in data collection and data labeling.

5.4 Attention-based Methods

Adding attention layers to this job not only didn't assist us, but it actually hurt our models' performance. RoBERTa F1-score dropped to 0.2959 thanks to the attention layer. The model with the attention layer earned an F1-score of 0.2145 for LSTM. The F1-score of BLSTM with attention layer was 0.2336.

5.5 Google's T5

Based on the hyperparameters listed in the methods section, our F1-score for this model is 0.4038, however we feel that by increasing the tokenization max length, increasing the epoch size, and utilizing the t5-large pretrained model, we may get even better results.

6 Conclusion

We discussed and compared numerous sarcasm detection algorithms in this work. We looked at the problem from numerous angles and reported our findings using each model. The F1-score of our best system, an ensemble model, was 0.414. All of the result are reported in Table 3.

Acknowledgements

We would like to convey our heartfelt gratitude to Prof. Yadollah Yaghoobzadeh of the University of Tehran, who provided us with invaluable advice during our research. We would also like to thank Ali Edalat from the University of Tehran, who provided us with the initial proposals for resolving the dataset's imbalance problem.

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.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116.

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*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586.

Xu Guo, Boyang Li, Han Yu, and Chunyan Miao. 2021. Latent-optimized adversarial neural transfer for sarcasm detection. *arXiv preprint arXiv:2104.09261*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sandeepa Kannangara. 2018. Mining twitter for fine-grained political opinion polarity classification, ideology detection and sarcasm detection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 751–752.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Stephanie Lukin and Marilyn Walker. 2017. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. *arXiv preprint arXiv:1708.08572*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Rishabh Misra and Prahal Arora. 2019. Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*.

Silviu Oprea and Walid Magdy. 2019. isarcasm: A dataset of intended sarcasm. *arXiv preprint arXiv:1911.03123*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Anwar Ur Rehman, Ahmad Kamran Malik, Basit Raza, and Waqar Ali. 2019. A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis. *Multimedia Tools and Applications*, 78(18):26597–26613.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 704–714.

Piyoros Tungthamthiti, Kiyoaki Shirai, and Masnizah Mohd. 2016. Recognition of sarcasm in microblogging based on sentiment analysis and coherence identification. *Journal of Natural Language Processing*, 23(5):383–405.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tony Veale, F Amílcar Cardoso, and Rafael Pérez y Pérez. 2019. Systematizing creativity: A computational view. In *Computational Creativity*, pages 1–19. Springer.

Marilyn Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.