I2C at SemEval-2022 Task 6: Intended Sarcasm Detection on Social Networks with Deep Learning

Pablo González Díaz
Escuela Técnica Superior de Ingeniería
Universidad de Huelva (Spain)
pablo.gonzalez682@alu.uhu.es

Pablo Cordón Hidalgo
Escuela Técnica Superior de Ingeniería
Universidad de Huelva (Spain)
pablo.gonzalez682@alu.uhu.es

Jacinto Mata Vázquez
Escuela Técnica Superior de Ingeniería
Universidad de Huelva (Spain)
mata@uhu.es

Victoria Pachón Álvarez
Escuela Técnica Superior de Ingeniería
Universidad de Huelva (Spain)
vpachon@uhu.es

Abstract

In this paper we present our approach and system description on iSarcasmEval: a SemEval task for intended sarcasm detection on social networks. This derives from our participation in *SubTask A: Given a text, determine whether it is sarcastic or non-sarcastic*. In our approach to complete the task, a comparison of several machine learning and deep learning algorithms using two datasets was conducted. The model which obtained the highest values of F1-score was a BERT-base-cased model. With this one, an F1-score of 0.2451 for the sarcastic class in the evaluation process was achieved. Finally, our team reached the 30th position.

1 Introduction

Sarcasm is a form of mockery intended to imply the opposite or to express displeasure. It can be classified as irony, satire, understatement, overstatement, rhetorical question and sarcasm itself. Social media platforms allow people to express themselves and speak about several different topics via text, emojis and multimedia. Many companies collect information about people's opinions to aid in marketing decision-making about their products.

Sarcasm detection is relevant in data analysis because it can be wrongly interpreted due to its nature, becoming potentially harmful to different computational systems. For example, if a computer system understands the sentence "*The only thing I got from college is a caffeine addiction*", its literal meaning would not be understood, as it refers to how stressful college can be.

Defining a phrase as sarcastic is a really challenging task to solve for the text mining community because there are many details that need to be considered such as the writer's personality and the sentence context. For instance, depending on the ideology of the person, "*Trump, that respectful man*" could be sarcastic or not.

In iSarcasmEval: Task 6, SubTask A (Abu Farha et al. 2022) participants had to decide if a particular tweet includes a sarcastic connotation or not.

2 Background

Two sample datasets were used to train the model.

The first one was the original dataset without any preprocessing. This dataset is composed of 3467 rows, with 2600 0-value rows (non-sarcastic) and 867 1-value rows (sarcastic).

The second sample dataset (Tweet – Rephrase) consists of a phrase and rephrase contrast in which the aim was to adequately learn the differences between a phrase with a sarcastic connotation and that same sentence expressing the same thing but in a literal way. This dataset is composed of 1734 rows: 867 1-value rows and 867 0-value rows, where these last ones are rephrases related to each tweet taken as a 1-value row.

So, these two datasets follow the same structure:

- tweet: *"See Brexit is going well"*
- sarcastic: "1"

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 877 - 880

In the second dataset, the column "rephrase" was added to the list of tweets. The rephrase example for the tweet below would be:

- tweet: "*Brexit really isn't going to plan.*"
- sarcastic: "0"

Table 1 shows some examples followed by both datasets.

The issue of sarcasm detection has been addressed by various authors in recent years. In the last several years, researchers have been working on a technique to analyze social media data in order to identify possible undisclosed information so it can be useful to assess new patterns and make important decisions through it (Shah and Shah 2021, 247-259). Sentiment analysis is also used for sarcasm detection (Suzuki et al. 2017), dividing a sentence into phrases and judging the situation and the sentiment separately.

In (Verma, Shukla, and Shukla 2021), the authors present a review of methodologies and techniques of sarcasm detection. They concluded that deep learning is the most common technique to identify sarcasm.

3 System overview

In this section, the models used to solve the sarcasm detection issue will be described.

Due to the nature of the problem, a decision was made not to carry out any preprocessing. The reason is that we think that sarcasm interpretation requires the original phrase to be considered in its entirety, so letter cases, emojis and punctuation were kept.

3.1 Models

A total of 30 models were used to train both datasets. In the first experiments, machine learning techniques such as LinearSVC, DecissionTrees and RandomForests using CountVectorizer and TFIDF (Pedregosa, Fabian and others 2011) were used. We also trained some deep learning techniques as LSTM Neural Networks (Hochreiter and Schmidhuber 1997) and Transformers (Vaswani et al. 2017).

For LSTM Neural Network we implemented the LSTM Simple Neural Network and we used the BERT model (Devlin et al. 2018) for Transformers.

In Table 2, the results of machine learning techniques mentioned before can be seen.

3.1.1 BERT: Bidirectional Encoder Representations from Transformers

BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning both the left and right contexts in all layers. As a result, the pretrained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question-answering and language inference, without substantial task-specific architecture modifications.

It uses Transformer, which is a simple network architecture based on attention mechanisms, dispensing with recurrence and convolutions entirely.

There are two ways to read a sentence for BERT. On one hand we have BERT-base-cased, in which BERT differences between capital letters and uppercase letters and, on the other hand, BERT-base-uncased, which will take the sentence and make all single letters uncased. As mentioned above, we did not miniscule the text of the tweets

Tweet	sarcastic
"Nice to be compared to a brick wall"	1
"Not happy i have been compared to a brick wall"	0
"Social Care for the young is basically a bath board and bed rest and your done"	1
"Social care for the young is non existent"	0
"I've been doing physics for 40 minutes I think I deserve a break"	1
"I'm never going to get it all done at this rate"	0

Table 1: Tweet examples

Dataset	Model		Metrics	
			Accuracy	F1-Score (class 1)
Original dataset	CountVectorizer	LinearSVC	0.68	0.28
		DecissionTree	0.74	0.10
		RandomForest	0.74	-
	TFIDF	LinearSVC	0.71	0.22
		DecissionTree	0.74	0.10
		RandomForest	0.74	-
Tweet – Rephrase	CountVectorizer	LinearSVC	0.60	0.61
		DecissionTree	0.56	0.37
		RandomForest	0.65	0.61
	TFIDF	LinearSVC	0.58	0.60
		DecissionTree	0.57	0.37
		RandomForest	0.61	0.61

Table 2: Results obtained in training phase using machine learning techniques

to preserve the sarcastic meaning of the sentence. Therefore, the BERT-base-cased was used.

The BERT model was trained with a batch size of 32 instances and 5 epochs by using a training/validation split.

3.1.2 LSTM Simple Neural Network

Long Short-Term Memory (LSTM) is a well-known recurrent neural network architecture. The most important feature of recurrent networks is their ability to persist introduction loops information in its diagram, so it can remember previous states and use this information in subsequent stages.

In order to find better results, a pretrained word vector with a size of 200d was added to our model. GloVe (Pennington, Socher, and Manning 2014) is a global log-bilinear regression model for the unsupervised learning of word representations that outperforms other models on word analogy, word similarity and named entity recognition tasks.

For this approach, a batch size of 32 instances, and 10 epochs was used. Besides, we implemented an early stopping to improve the performance.

4 Experimental setup

Many libraries were imported to develop the experiments. Some of them are: “NumPy” (Harris et al. 2020)”, a fundamental package for scientific computing; “spaCy” (Honnibal and Montani 2017), a library for advanced natural language processing; “NLTK” (Loper and Bird 2002), a tool-kit to work with human language information; “Keras” (Chollet 2015), a neural-network library; “scikit-learn”, which contains simple and efficient tools for predictive data analysis and “Pandas” (McKinney 2010), used for manipulating data.

For all the experiments, the datasets were split using 80% to train and 20% to test. We use a stratify approach.

Finally, when the test dataset was released, the whole train dataset was used to train the final model through BERT-base-cased, which was the best algorithm to complete the task.

The metrics obtained to evaluate our model are accuracy and F1-score.

5 Results

Table 3 shows the results obtained in our experiments during the training phase. The algorithm that reached the best results was BERT-base-cased with the use of the Tweet - Rephrase dataset, obtaining a score of 0.86 accuracy and F1-score.

Results shown in the leaderboard with the test dataset yielded a 0.2451 F1-score. Finally, our team reached the 30th position.

6 Conclusions

In this paper our approach to solve Task 6 (iSarcasmEval) - SubTask A: Given a text,

Dataset	Model	Metrics	
		Accuracy	F1-Score (class 1)
Original dataset	LSTM Simple Neural Network	0.65	0.38
	BERT-base-cased	0.72	0.43
	BERT-base-uncased	0.69	0.39
Tweet – Rephrase	LSTM Simple Neural Network	0.65	0.62
	BERT-base-cased	**0.86**	**0.86**
	BERT-base-uncased	0.84	0.84

Table 3: Results obtained in training phase using deep learning techniques

determine whether it is sarcastic or non-sarcastic, in English, has been described.

The main idea was to check models of deep learning algorithms such as BERT or LSTM Neural Network trained with the dataset Tweet – Rephrase. After training and analyzing each model and comparing deep learning with machine learning techniques, BERT-base-cased model obtained higher results F1-Score and accuracy.

In the future, a BERT model will be trained using a larger corpus and more innovative techniques such as the application of sentiment analysis in broken-down sentences.

References

Abu Farha, Ibrahim, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. "SemEval-2022 Task 6: iSarcasmEval Intended Sarcasm Detection in English and Arabic."Association for Computational Linguistics, .

Chollet, Francois and others. 2015. "Keras." . https://github.com/fchollet/keras.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." . https://arxiv.org/abs/1810.04805.

Harris, Charles R., K. Jarrod Millman, van der Walt, Stéfan J., Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, et al. 2020. "Array Programming with NumPy." Nature 585: 357–362. doi:10.1038/s41586-020-2649-2.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." Neural Computation 9 (8): 1735-1780. doi:10.1162/neco.1997.9.8.1735. https://doi.org/10.1162/neco.1997.9.8.1735.

Honnibal, Matthew and Ines Montani. "spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing.".

Loper, Edward and Steven Bird. 2002. "NLTK: The Natural Language Toolkit." Philadelphia, Pennsylvania, Association for Computational Linguistics, . doi:10.3115/1118108.1118117. https://doi.org/10.3115/1118108.1118117.

McKinney, Wes and others. 2010. "Data Structures for Statistical Computing in Python."Austin, TX, .

Pedregosa, Fabian and Varoquaux, Ga\el and Gramfort, Alexandre and Michel, Vincent and Thirion, Bertrand and Grisel, Olivier and Blondel, Mathieu and Prettenhofer, Peter and Weiss, Ron and Dubourg, Vincent and others. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12 (Oct): 2825-2830.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "Glove: Global Vectors for Word Representation."01. doi:10.3115/v1/D14-1162.

Shah, Bhumi and Margil Shah. 2021. "A Survey on Machine Learning and Deep Learning Based Approaches for Sarcasm Identification in Social Media."Springer Singapore, .

Suzuki, Shota, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara, and Akihiko Ohsuga. 2017. "Sarcasm Detection Method to Improve Review Analysis.".

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. *Attention is all You Need.*

Verma, Palak, Neha Shukla, and A. P. Shukla. 2021. "Techniques of Sarcasm Detection: A Review.". doi:10.1109/ICACITE51222.2021.9404585.