Profiling Irony Speech Spreaders on Social Networks Using Deep Cleaning and BERT

Leila Hazrati[a], Alireza Sokhandan[a] and Leili Farzinvash[a]

[a] *Computer Eng. Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran*

Abstract

With irony, language is employed figuratively and subtly to mean the opposite of what is stated. In the case of sarcasm, a more aggressive type of irony, the intent is to mock or scorn a victim without excluding the possibility to hurt. Stereotypes are often used, especially in discussions about controversial issues such as immigration, sexism, and misogyny. Regarding PAN's open submission toward tackling this issue, we use BERT (bidirectional encoder representations from transformers) as a way to identify ironic and sarcastic phrases from genuine ones in Twitter posts. Since the goal is to detect irony in texts published on social media, and usually social media users have a different writing style and their texts contain a variety of nonstandard language expressions, the input texts are deep cleaned before feeding into the BERT network. The experimental results show a significant improvement in the accuracy and training loss ratio of the BERT network by applying deep cleaning to input texts. Thus, we achieved up to 98.5 percent accuracy with the proposed method.

Keywords 1

Irony Detection, Author Profiling, Stereotypes, BERT, Text Cleaning

1. Introduction

A common definition of verbal irony is saying things opposite to what is meant [1]. Many studies have diverse opinions regarding sarcasm and irony being different phenomena [2] or being the same [3]. Irony as a literary technique is a linguistic device used in social networks such as Twitter to intent an idea while articulating an opposite expression. In this work, we don't differentiate between irony and sarcasm and intent to perform irony speech spreader identification on social media based on their post, to address the PAN-2022 task (IROSTERO 2022) which aims to profile irony and stereotype spreaders on Twitter. [4,5]

In the proposed method, we use a transformer model, called BERT [6] as the feature extractor. BERT is designed to help computers understand the meaning of ambiguous language in a text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned for any task by using a related dataset. By adding two dense layers on top of the BERT model, the input text data is binary classified as irony or not irony.

Approaches based on neural networks (word embedding) treat any input characters including special characters such as ",", ".", "!", "?", "#", and username mentions ("@") as a regular word [7, 8]. These approaches often do not perform data cleaning [7,9], considering that the network itself would solve the related problems [10]. In some works, where word embedding is used, and especially in cases where there are not enough training samples, the use of basic data cleaning significantly improves the feature representation [11,12,13]. In this work, considering that the input is the text posted on social media and in such texts, grammatical or linguistic principles are usually not fully

[1]CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
EMAIL: leilahazrati.75@gmail.com; a.sokhandan@tabrizu.ac.ir; l.farzinvash@tabrizu.ac.ir
ORCID: 0000-0003-0318-2156 ; 0000-0001-6505-8957 ; 0000-0002-2369-9833

observed, and from the other side, the database provided by the organizer (described in Section 4.1) has a small number of samples, a deep and targeted data cleaning such as URL filtering, stop words removal, removing punctuations, etc. is performed to increase the classification accuracy. Based on the experimental results (available on the TIRA [14] platform), the proposed method obtained 98.50% accuracy for detecting irony speaking on the provided dataset.

The rest of the paper is organized as follows: In Section 2, we present the related work on the irony detection field; in Section 3, we describe the methodology used for classification; in Section 4, the results of our experiments are presented, and finally Section 5, includes conclusion and future work.

2. Related Work

With the merge of the social media era and the contribution of users on online platforms, a vast amount of data for emotional human linguistic behavior analysis can be retrieved as datasets mined from these platforms, to be specific Twitter. Because Twitter users express their feelings and opinions on social networks with frequent irony [15], thus Twitter is a platform of choice due to its users posting their thoughts in stereotypical and ironic ways more than usual. Several approaches to irony and sarcasm detection have been developed. Some studies have used feature sets of the text to classify the text as ironic or not [16]. In [17], Transformers architecture is used to contextualize pre-trained word embedding, they contextualize Word2Vec word embedding which is the authors' profile vector. They used BERT in transformer-based contextualization of pre-trained word embedding for irony detection on Twitter. In [18], BERT is used for irony detection in the Portuguese language, and BERTs' output is considered as the authors' profile descriptor. In this work, the main objective is feature selection to choose a subset used in the classification process. It mentioned that the text information comprises features that are redundant and irrelevant, and the redundant features have no contribution to separating the classes from each other. Research in [19] is based on using various neural network models, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Baseline Convolutional Neural Networks (CNN) in an ensemble model to detect sarcasm on the internet, they used word embedding which is a method of representing words in the numeric vector form. It learns a divided representation of a vocabulary from a collection of text. It possesses the ability to divulge several unknown relationships between the words, the idea of the word embedding is that if there is a user in the training data with a certain personality, and they happen to make sarcastic tweets, then when it gets new data and there is a new user that has a similar style and therefore similar embedding to the previous user, without looking at the new user's tweet, it can predict if this user will be sarcastic or not, just by looking at the similarity of the embedding. This technique enables sharing the representation across words which helps in creating a more stable representation of words. [20] Presents an Intelligent ML-based sarcasm detection and classification (IMLB-SDC) technique. This technique encompasses different stages such as preprocessing, TF-IDF-based feature engineering, SVM-based classification, and PSO-based parameter tuning, and two Feature selection approaches are utilized, namely chi-square and information gain. The IMLB-SDC model relay on the Support vector machine as a classification model; and [21] surveys the current state-of-the-art and presents strong baselines for sarcasm detection based on BERT pre-trained language models. These methods only relay on the power of transformer models, and the necessity of data preparation is not well addressed.

3. Proposed Method

The proposed method consists of three main steps: clearing the input text, feature extraction, and classification, in these three steps, receiving the tweets of a user, determines whether that user is an ironic speaker or not. The overall procedure of the proposed method is shown in Fig 1.

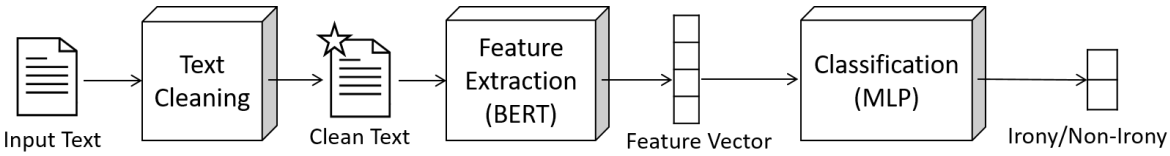

Figure 1: Overall procedure of the proposed method

3.1. Text Cleaning

Considering the irony speech detection task, in the first step we apply a deep and targeted text cleaning to the input text including the following items:

- Unescaping HTML codes, such as &amp with correspondent characters
- Replacing the emojis with equivalent text (Table 1)
- Removing the URLs
- Replacing digits with "number"
- Removing none English phrases
- Removing stop words
- Removing words such as "hashtag", "user"
- Removing punctuations and any control characters
- Separating words that were stuck together by characters such as "/" and "\"

For replacing the emojis with equivalent text we use the *demojize* function from emoji's python module. [22] in this module, the entire set of emoji codes as defined by the Unicode consortium is supported. We have to pass the emoji as an argument inside the *demojize* function, and its CLDR (Common Locale Data Repository) short name will be returned, which is a meaningful description of that emoji in the form of a few words.

Table 1
Replacing emojis with equivalent text

Tweets with emojis	Replaced emoji with equivalent text
Good morning everyone! 😊	Good morning everyone! Smiling face with open hands
Good enough. 😡	Good enough. Pouting face

3.2. Feature Extraction & Classification

In this work, we benefit from the BERT base model for binary classification. This model contains an encoder with 12 transformer blocks, 12 self-attention heads, and a flat layer with a size of 768. BERT takes an input sequence of no more than 512 tokens and outputs the representation of the sequence. The sequence has one or two segments. The first token of the sequence is always [CLS] which contains the special classification embedding and another special token [SEP] is used for separating segments. For text classification tasks, BERT takes the final hidden state of the first token [CLS] as the representation of the whole sequence. To use BERT or any other model, it must be further pre-trained based on the task at hand. There are three approaches:

1. Further pre-train the BERT on the training data for the target task, called within-task pre-training
2. Using a further pre-trained model of the other task from the same domain of the target task is called in-domain pre-training
3. Cross-domain pre-training, in which the further pre-trained model is obtained from a task with different domains than the target task [23]

We use the first approach for our work. In this case, the error is back-propagated through the entire architecture and the pre-trained weights of the model are updated based on the new training dataset.

The output of the BERT model represents the feature vector of the input text and is passed into the Multi-Layer Perceptron (MLP) network to get the conclusion, of which, is the author of the input text is an ironic speaker or not. The MLP network which acts as a classification model consists of one input layer with the size of 768, two hidden layers containing 512 and 128 neurons respectively, with RELU (rectified linear unit) activation function, and a softmax output layer.

The pre-trained BERT model is followed by the classifier i.e. three dense layers, which together form a deep neural network, trained simultaneously, using the ADAM algorithm. In this process by using the loss function calculation by cross-entropy method, we update the MLP parameters which are the weights between dense layers, and simultaneously fine-tune the BERT's network parameters (the weights of embedding, attentions, and encoder layers which are almost 110M value) corresponding with given PAN's task.

In the proposed method, all the cleaned tweets of an author are concatenated and given to the BERT model, which gives us a numerical descriptor with 768 dimensions, considered as the identifier for the author's profile.

4. Experiments & Results

In this section, firstly, we introduce the dataset that is employed. Then, we report and discuss the experimental results of the proposed method.

4.1. Dataset

The given data set provided by PAN [24] (shown in Table 2), consists of 420 authors (Twitter users) having 200 tweets each. The grand truth of the dataset is a binary flag that determines whether the author's text is Ironic or not. The dataset is symmetric, containing 210 ironic speakers and 210 non-ironic speakers**.**

Table 2
Dataset schema

Name	Samples	Task	Label	Year
pan22-author-profiling-training-2022-03-29	420	Binary	I (210 Sample) NI (210 Sample)	2022

4.2. Experiments

We implement and review our proposed model in three ways, and the experimental results of these approaches are reported in Fig. 2, in terms of loss value (calculated by the cross-entropy method [25]), and Table 3 in terms of accuracy.

1. Text classification without any data cleaning
2. Text classification with basic data cleaning: In this method, we only use basic preprocessing, which is not involve replacing emojis, deleting numbers, removing non-English words, and separating glue words.
3. Text classification with deep data cleaning: This method includes all the cleaning steps described in Section 3

As shown in Fig. 2, with aid of deep data cleaning, in our model, validation and training loss, are significantly reduced, compared to the 1st and 2nd approaches; and the amount of training loss tended toward zero. Generally, with a reduction of loss values, we looking for gaining accuracy, according to Table 2, because of using deep preprocessing, in comparison to other modes, our proposed model's accuracy significantly increased reaching 98%.

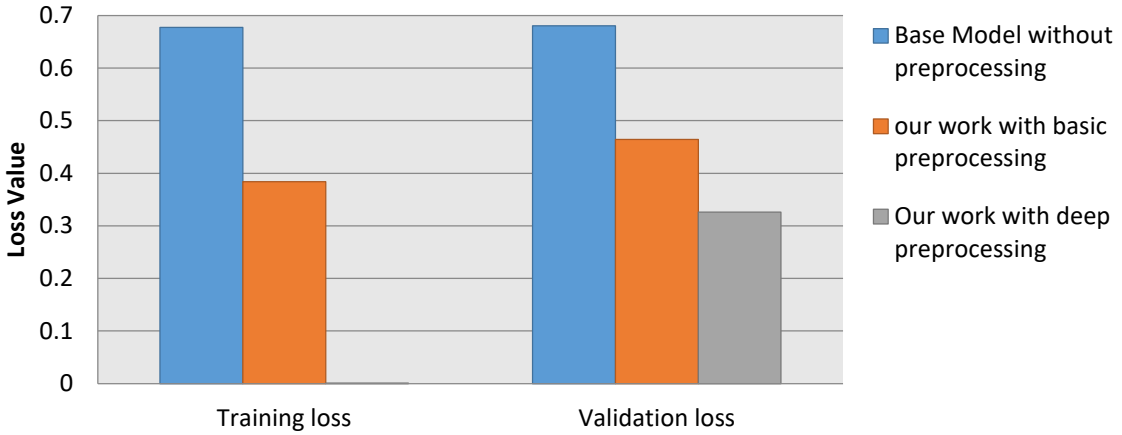

Figure 2: Comparison of our work versus base and basic preprocessed models

Table 3
Comparison between our work and BERT base model and basic preprocessed model

method	Accuracy	Precision	Recall	F1-score
Without text cleaning	0.6240	0.5816	0.8636	0.6950
Basic text cleaning	0.8345	0.8571	0.7741	0.8134
Deep text cleaning	**0.9849**	**0.9850**	**0.9850**	**0.9850**

From the obtained results, it can be seen that suitable cleaning has a significant effect on the results and accuracy of our model in the feature extraction and classification, especially when our data involves social media text that contains a large variety of nonstandard language expressions. By comparing the results of the second (normal cleaning) and third (deep cleaning) approaches, we come to the conclusion that targeted cleaning of data can be useful in diagnosing irony, because emoji is a medium for people to express emotions and their personalities, and conveyed the feelings of the writers more "authentically". Also from this comparison, it can be seen that the numbers are not an important factor in the discussion of ironic speech and their removal can be very effective in increasing the accuracy of irony detection. Results show that proportional BERT final layer size adjustment can increase the classification accuracy. The size of these layers should be commensurate with the number of data available, and since the task's data was limited, we consider fewer neurons for these layers. If more samples were available, we can use more neurons and layers to fine-tune BERT and improve model accuracy to cover a variety of states in the dataset.

5. Conclusion & Future Work

In this work, our goal was to identify users and writers who spoke sarcastically. To address this issue, we employed a transformer network as the main base, by performing a deep and targeted cleaning, when appropriate data with proper adjustment is injected into the BERT model, it returns accurate results. The experimental results show that compared to basic cleaning, targeted cleaning

significantly increased the accuracy, so it is possible to generalize this issue to the other sentimental analysis tasks to achieve higher accuracy.

In the future, we focus on preprocessing phase and specifically will work on identifying slang words and examine the effect of manipulating them on the accuracy of irony detection

6. References

[1] Carmen, Curcó. Irony: Negation, Echo, and Metarepresentation. Irony in Language and Thought, (2007): pp. 269–296, URL: https://doi.org/10.1016/S0024-3841(99)00041-8.

[2] H. P. Grice. Logic and conversation. Speech acts, ed. by peter cole and jerry morgan, (1975): 41-58, URL: https://doi.org/10.1163/9789004368811_003.

[3] A. Reyes, P.Rosso, and T.Veale. A multidimensional approach for detecting irony in Twitter. Language Resources and Evaluation, (2013): 47(1): 239–268, URL: https://doi.org/10.1007/s10579-012-9196-x.

[4] R. Ortega-Bueno, B. Chulvi, F. Rangeland, P. Rosso and E. Fersini. Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022. In: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org

[5] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska and E. Zangerle, Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection, vol.13390, 2022.

[6] J. Devlin, M. Chang, K.Lee, and K.Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Computation and Language, 2019, URL: https://doi.org/10.48550/arXiv.1810.04805.

[7] Q. Le and T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning (ICML ’14), Beijing, China, 2014, pp. 1188–1196, URL: https://doi.org/10.48550/arXiv.1405.4053 .

[8] I. Brigadir, D. Greene, and P. Cunningham, Adaptive representations for tracking breaking news on Twitter, Information Retrieval (cs.IR); Neural and Evolutionary Computing, 2014, URL: https://doi.org/10.48550/arXiv.1403.2923.

[9] R. Socher, C. C. Y. Lin, C. D. Manning, and A. Y. Ng, Parsing natural scenes and natural language with recursive neural networks, in: Proceedings of the 28th International Conference on Machine Learning (ICML ’11), Bellevue, Wash, USA, 2011, pp. 129–136.

[10] H. G. Adorno, I. Markov, G. Sidorov, Improving Feature Representation Based on a Neural Network for Author Profiling in Social Media Texts, 2016, doi:10.1155/2016/1638936

[11] C. Yan, F. Zhang, and L. Huang, DRWS: a model for learning distributed representations for words and sentences, in: PRICAI 2014: Trends in Artificial Intelligence, D. N. Pham and S. B. Park, Eds., vol. 8862 of Lecture Notes in Computer Science, pp. 196–207, Springer, 2014, doi: 10.1007/978-3-319-13560-1_16

[12] V. K. R. Sridhar, Unsupervised text normalization using distributed representations of words and phrases, in: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing (NAACL ’15), Association for Computational Linguistics, Denver, Colo, USA, 2015, pp. 8–16, doi: 10.3115/v1/W15-1502.

[13] F. Jiang, Y. Liu, H. Luan, M. Zhang, and S. Ma, Microblog sentiment analysis with emoticon space model, Social Media Processing, Springer, Berlin, Germany, 2014, pp. 76–87, doi: 10.1007/s11390-015-1587-1.

[14] M. Potthast, T. Gollub, M. Wiegmann and Benno Stein, TIRA Integrated Research Architecture, Information Retrieval Evaluation in a Changing World, Springer, Berlin, Germany, 2019, doi: 10.1007/978-3-030-22948-1_5

[15] Amir, S., Wallace, B. C., Lyu, H., and Silva, P. C. M. J., Modelling context with user embeddings for sarcasm detection in social media, 2016, doi: arxiv:1607.00976.

[16] A.Reyes, P.Rosso, and T.Veale, A multidimensional approach for detecting irony in Twitter, Language resources and evaluation, (2013): 47(1): 239–268, doi: 10.1007/s10579-012-9196-x.

[17] G.Á. José, F. H. Lluís, P.Ferran, Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter, 2020, doi: 10.1016/j.ipm.2020.102262
[18] J.Shengyi, C.Chuwei, L.Nankai, Z.Chen, and J.Chen, Irony Detection in the Portuguese Language using BERT, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN), Málaga, Spain, 2021.
[19] P.Goel, R.Jain, A.Nayyar, Sh.Singhal1 and M.Srivastava1, Sarcasm detection using deep learning and ensemble learning, 2022, doi: 10.1007/s11042-022-12930-z
[20] D.Vinoth, P. Prabhavathy, An intelligent machine learning-based sarcasm detection and classification model on social networks, 2022, DOI: 10.1007/s11227-022-04312-x
[21] E.Savini, C.Caragea, Intermediate-Task Transfer Learning with BERT for Sarcasm Detection, 2022, doi: 10.3390/math10050844
[22] T.Kim, K.Wurster, T. Jalilov, Emoji for Python v1.7.0, URL: https://pypi.org/project/emoji/
[23] C. Sun, X. Qiu, Y. Xu, Xuanjing H., How to Fine-Tune BERT for Text Classification?, LNAI 11856, 2019, pp. 194–206, URL: https://doi.org/10.1007/978-3-030-32381-3_16.
[24] Reynier O., Bueno, Berta C., Fransisco R., Paolo R, and Elisabetta Fersini, Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO), PAN 22 Author Profiling, 2022, URL: https://doi.org/10.5281/zenodo.6397037.
[25] De B., P. T., Kroese, D.P, Mannor, S. and Rubinstein and R.Y. A Tutorial on the Cross-Entropy Method, Annals of Operations Research, (2005): 134 (1): pp. 19–67, URL: https://doi.org/10.1007/s10479-005-5724-z.