Irony Detection Using Transformers

Abhishek Agrawal
Dept. of Computer Engineering
Delhi Technological University
New Delhi, India
abhishekagrawalsr@gmail.com

Abhishek Kumar Jha
Dept. of Computer Engineering
Delhi Technological University
New Delhi, India
abhishekjha7798@gmail.com

Ashish Jaiswal
Dept. of Computer Engineering
Delhi Technological University
New Delhi, India
jaiswalashish1403@gmail.com

Dr. Vinod Kumar
Dept. of Computer Engineering
Delhi Technological University
New Delhi, India
vinod_k@dtu.ac.in

***Abstract*— With the ever-expanding social net, the use cases of irony detection and classification is also exponentially increasing. With this work, we take Task 3 of SemEval-2018 as our problem statement which further has two tasks. We intend to first determine if a given tweet is ironic or not (Task A) and then classify the tweets into four classes viz. non-ironic, verbal irony with contrast, verbal irony without contrast and situational irony (Task B). Existing papers have mainly exploited the lexical features of tweets using supervised machine learning. Here, we have proposed two NLP Transformer models viz. BERT (Bidirectional Encoder Representations from Transformers) and XLNets to classify tweets and have also compared our results to that of past papers. Using BERT, we have achieved F1 scores of 0.70 and 0.75 and using XLNets 0.74 and 0.59 for Task A and Task B respectively.**

Keywords*— *BERT, XLNets, transfer learning, classification, irony detection

I. Introduction

The evolution of social networks has promoted the use of sarcastic and ironic expressions in public posts. This is done to present one's views in a concise, effective and attractive manner. Over time, irony detection and classification have found a lot of use cases in real-world scenarios such as sentiment analysis, detecting online harassment, author profiling, recording and analyzing someone's views.

Our job here is to model Task 3 of SemEval-2018 [1] where we first determine if a given tweet is ironic or not and then classify them into four categories. "I really love catching a cold again and again" and "This quilt is as soft as a brick" are examples of verbal irony realized through polarity contrast. On the other hand, "Keeping the environment clean, that's what he wants. #irony" is an example of verbal irony without polarity contrast and "Breaking news!! Policeman got robbed yesterday night." is an example of situational irony. The fourth class contains tweets that are not ironic.

Existing works on irony detection mainly focussed on the linguistic details of the English language. But not much work has been done using Transformer models like BERT and XLNets. This is quite the opposite of what has been done in earlier papers. Instead of reading text sequentially (either from left to right or from right to left), these autoencoder lingustic models are trained bidirectionally to understand the context of a word in the text more accurately. XLNET is different from BERT as the former uses the permutation method and the latter uses the MASK method. In this paper, we have used these two models to provide a better solution to irony detection and classification. Here, we have considered four metrics viz. accuracy, precision, recall and f1 score which were used to compare our results to that of the past papers.

Our primary contributions to this problem include:

- Training different NLP Transformer models like BERT and XLNets for the purpose of detecting irony in tweets and further classifying them.
- Analyzing efficacy of transfer learning in better understanding and detecting the types of irony present in the given tweets.

II. Related Work

Irony detection and classification have been an area of growing interest in recent years because of its importance in sentiment analysis. In this part, we briefly discuss the approaches taken in previous researches on the same problem. THUNGN Reference [2] proposed a densely-connected LSTM network employing a multitask learning strategy. It is a sequential model in which every layer takes all the results from preceding layers as its input. The output produced from the final layer is used for the classification task. They achieved the best result for task 1 and did fairly well in task 2. Irony Magnet Reference [3] used a Siamese neural net consisting of two subnetworks. Each of these subnetworks uses an LSTM combined with an embedding layer, the layer being initialized with Glove word embedding vectors. WLV Reference [4] proposed using an ensemble soft voting classifier with LR (Logistic Regression) and SVM as constituent models. They created a feature set using semantic and sentiment features. They use this feature set along with dense vector representations for both tasks 1 and 2. NTUA-SLP Reference [5] used two independent Bi-LSTMs. They also combined a self-attention mechanism to detect the most significant words for classification tasks. They initialized the embedding layer with word2vec embeddings.

ELiRF-UPV Reference [6] uses a combination of LSTMs and Convolutional Neural Networks (CNN) for both subtasks. NIHRIO Reference [7] uses a Multilayer

978-1-7281-7106-7/20/$31.00 ©2020 IEEE
DOI 10.1109/CDS49703.2020.00040

Perceptron for both tasks. The input layer represents the text using a feature vector, which contains syntactic, polarity, lexical and semantic feature representations. The hidden layers choose the most significant attributes for the given task. These attributes are then inputted to a softmax layer for classification.

KLUEnicorn Reference [8] proposed a system using naive Bayes classifier as the main underlying model. They made use of various adverb categories and named entities along with the semantic and lexical features to generate word embeddings. Reference [9] provides a comprehensive review of the various supervised classification algorithms for irony detection. They evaluate and compare the results of the following models - Voted Perceptron, Radial Basis Function Networks (RBF), Fuzzy Lattice Reasoning (FLR), Randomizable Filtered Classifier (RFC), Isolation Forest, Logistic Model Tree (LMT), Bayesian Network (BayesNet), OneR, Stochastic Gradient Descent (SGD), IBk, Multi-Layer Perceptron (MLP) and Bagging. Among these, RFC, Voted Perceptron and IBk were found to produce best results for all evaluation metrics.

In this paper, we focus mainly on the transformer models for text classification, namely BERT and XLNet.These transformer models are found to achieve better results than most state-of-the-art models. The next section discusses in detail our approaches.

III. Our Methodology

Here, our aim is to propose a model to solve the irony detection binary and multi-class classification problem of SemEval. After the research done on the previous methods proposed to solve this problem, we came up with an innovative solution to use Transformer models. The following section gives a detailed description of the steps taken for the construction of these experiments.

A. Preprocessing

The dataset was preprocessed to remove the hashtag symbol from the tweets. The tagging of users in the tweet such as '@someuser' has been replaced with the user instead, for the better processing of data in the model. Similarly, the web links have been replaced with a link in the text. The punctuations have been intentionally kept in the tweets as the Transformer models create improved contextual embeddings in the background of the model, in the presence of punctuations in the sentences. For a better analysis of the data, we replaced emojis and emoticons from the tweets with its description using python library emot.

A brief explanation of the models and their implementation has been done in this subsection. We made use of the HuggingFace [10] python library which provides state of the art implementations of the general-purpose architectures such as BERT and XLNets.

B. BERT

BERT (Bi-directional Encoder Representations from Transformers) reference [12] is a deep bidirectional unsupervised language representation model trained on a huge plain-text Wikipedia corpus.

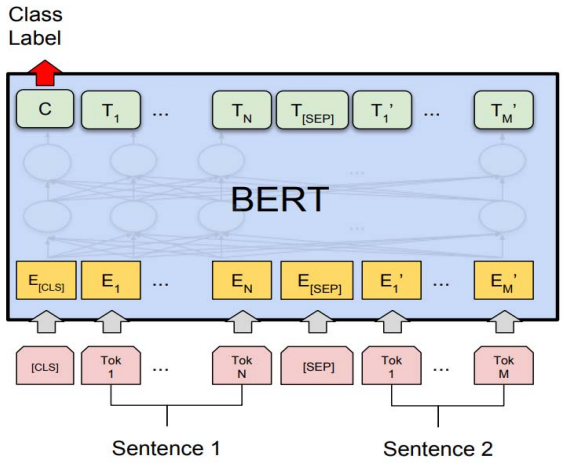

Fig. 1. BERT Model

TABLE I. HYPERPARAMETERS OF MODELS

Task	Model	Hyperparameters
A	BERT-base	lr=3e-5, bs=32, epochs=5, Steps per epoch=7 optimizer=Adam
B	BERT-base	lr=3e-5, bs=32, epochs=5, Steps per epoch=7 optimizer=Adam
A	XLNet	lr=3e-5, bs=32, epochs=5, Steps per epoch=7 optimizer=Adam
B	XLNet	lr=3e-5, bs=32, epochs=5, Steps per epoch=7 optimizer=Adam

lr= Learning Rate

bs= Batch-Size

BERT is built on top of the Transformer encoder stack and the semi-supervised sequence learning approaches of NLP. It takes a sentence as input and understands the contextual relationship between words in a sentence, then the concept of attention mechanism is used wherein a mask randomly selects a word from a sentence and predicts its context based on the words in the vicinity. Bert has two model sizes Base and Large. The base model has 12 transformer blocks in it whereas the Large model has 24 transformer blocks in it. Each transformer block has 768 and 1024 feed-forward hidden layer units respectively and attention layer with 12 and 16 units respectively.

For both our problem tasks, we used the Tensorflow based implementation of Bert-base model with the help of the HuggingFace library and fine-tuned the model on our dataset.

The table below shows the hyperparameters chosen for the fine-tuning of the BERT-base and XLNet models.

C. XLNets

XLNet [13] is a generalized auto-regressive language model which is built on top of Transformer-XL. Transformer-XL makes use of a recurrence mechanism that captures long term dependencies and avoids context fragmentation in longer sentences. It also implements relative positional encoding to the transformer architecture so that the relative distance of the context from the current word is encoded continuously at each attention module. To capture the bi-directional context of sentences XLNets make use of Permutation Language Modelling in which a token is predicted based on all possible permutations of words in a sentence. The main objective of PLM is to maximize the log-likelihood of the target token word over all the words in a sentence. Thus we do not need a mask to capture the context in XLNets as compared to Bert.

We made use of the HuggingFace library for implementing the XLNet model and fine-tuned the model on our dataset. The table [1] states the hyperparameters chosen by us for fine-tuning the XLNet model for both the classification tasks.

IV. Evaluation

In this section, a detailed overview of the dataset made available by the 'SemEval-2018 Task 3: Irony detection in English tweets' challenge is provided. We then present the various metrics used to evaluate the performance of our models. The section finally concludes with the details of several hyperparameters used and the task results obtained.

A. Dataset for SubTask A

Fig. 2 gives a pictorial representation of the composition of the dataset for subtask A. The training data for this subtask contains 3817 English tweets. Out of these 3817 tweets, 1901 are ironic and 1916 are non-ironic. The test set consists of 784 tweets (311 ironic and 473 non-ironic).

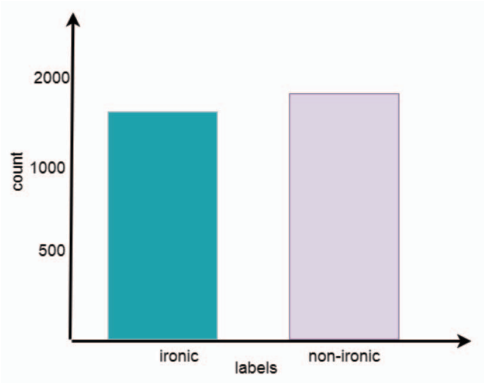

Fig. 2. Class Distribution of Dataset of SubTask A

B. Dataset for SubTask B

Fig 3 shows the distribution of different types of ironies in the training corpus for subtask B.

There are a total of 3817 tweets. The exact composition is as shown in table II.

The test set contains 784 samples (473 non-ironic, 164 ironic with polarity contrast, 85 situational ironic and 62 ironic without polarity contrast).

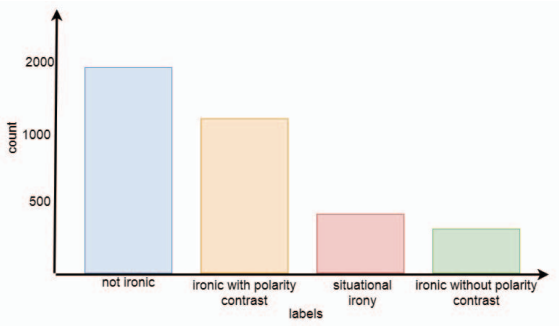

Fig. 3. Class Distribution of Dataset of SubTask B

TABLE II. DETAILED DISTRIBUTION OF THE DATASET

Class Label	No. of Samples
Ironic with polarity contrast	1383
Situational Irony	316
Ironic without polarity contrast	202
Not Ironic	1916

C. Evaluation Metrics

The evaluation metrics chosen for this SemEval challenge were accuracy, precision, recall, and F1 score. The performance of different models is compared using the F1 score.

The metrics are computed as follows:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (4)$$

Where,

TP= True Positive TN=True Negative
FP= False Positive FN= False Negatives

The following section shows our results evaluated on the above metrics.

D. Results

The table III shows the results of Task A in which binary classification of ironic and non-ironic tweets has been done using BERT and XLNet. We found out that the XLNet model performed better than BERT model and also surpassed the state of the art results for Task A.

TABLE III. EXPERIMENTAL RESULTS OF TASK A

Model	Epochs	Accuracy	Precision	Recall	F1 Score
BERT	3	0.66	0.75	0.66	0.70
XLNet	5	0.70	**0.79**	0.70	**0.74**
Semeval Reference [1]	-	**0.73**	0.63	**0.80**	0.70

The table IV shows the results of Task B which is a multi-class classification of various types of irony in tweets. We found out that the BERT model performed better than XLNet for the multi-class problem and is comparable with the state of the art results obtained in the SemEval Task 3.

The table IV shows the F1 Score for each subclass. The table V exhibits the detailed weighted average score of all the metrics for the two Transformer models.

TABLE IV. EXPERIMENTAL RESULTS OF TASK B

Model	Epochs	Not ironic	Ironic with polarity contrast	Situational irony	Ironic without polarity contrast
BERT	5	**0.86**	**0.72**	**0.73**	0.06
XLNet	10	0.71	0.56	0.35	0.07
Semeval Reference [1]	-	0.84	0.69	0.46	**0.23**

TABLE V. DETAILED RESULTS OF TASK B

Model	Epochs	Accuracy	Precision	Recall	F1 Score
BERT	5	**0.77**	**0.82**	**0.77**	**0.75**
XLNet	10	0.60	0.64	0.60	0.59

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a transformer based solution to Task 3 of SemEval-2018 of detecting irony in English Tweets. We were successful in using BERT and XLNets to understand the contexts of a word in a tweet more accurately and use these models for irony detection and classification. In future, we plan to include emojis to incorporate the sentiment aspect of the tweet which will improve the understanding of the ironic tweets. We can also use sampling techniques to further improve the multiclass classification model. Moreover, we also plan to use the BERT large model which has more number of hidden and attention layers thus more computationally intensive instead of using the current BERT base model for improving the results.

REFERENCES

[1] Van Hee, Cynthia, Els Lefever, and Véronique Hoste. "Semeval-2018 task 3: Irony detection in english tweets." Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.

[2] Wu, Chuhan, et al. "Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning." Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.

[3] Ghosh, Aniruddha, and Tony Veale. "Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media." Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.

[4] Rohanian, Omid, et al. "Wlv at semeval-2018 task 3: Dissecting tweets in search of irony." Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.

[5] Baziotis, Christos, et al. "Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns." arXiv preprint arXiv:1804.06659 (2018).

[6] González, José-Ángel, Lluís-F. Hurtado, and Ferran Pla. "ELiRF-UPV at SemEval-2018 Tasks 1 and 3: Affect and irony detection in tweets." Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.

[7] Vu, Thanh, et al. "NIHRIO at semeval-2018 task 3: A simple and accurate neural network model for irony detection in twitter." arXiv preprint arXiv:1804.00520 (2018).

[8] Dürlich, Luise. "KLUEnicorn at SemEval-2018 Task 3: A Naive Approach to Irony Detection." Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.

[9] Baloglu, Ulas Baran, Bilal Alatas, and Harun Bingol. "Assessment of Supervised Learning Algorithms for Irony Detection in Online Social Media." 2019 1st International Informatics and Software Engineering Conference (UBMYK). IEEE, 2019.

[10] Wolf, Thomas, et al. "Huggingface's transformers: State-of-the-art natural language processing." ArXiv, abs/1910.03771 (2019).

[11] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[12] Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pretraining for language understanding." Advances in neural information processing systems. 2019.