Contextualized Embedding based Approaches for Social Media-specific Sentiment Analysis

Harsh Sakhrani§
Pune Institute of Computer Technology
Pune, India
harshsakhrani26@gmail.com

Saloni Parekh§
Pune Institute of Computer Technology
Pune, India
saloniparekh1609@gmail.com

Pratik Ratadiya
vCreaTek Consulting Services Pvt. Ltd.
Pune, India
pratik.r@vcreatek.com

***Abstract*—Social media-specific Sentiment Analysis has a wide range of applications in various domains like Business Intelligence, Marketing, Politics and Psychology, to mention a few. Irony Detection and Emotion Recognition, two of Sentiment Analysis' significant pillars have become increasingly important as a result of the continued growth of social media. Previous approaches for the two tasks have yielded promising results, but have often relied on recurrence and pre-trained word-embedding ensembles. In this paper, we propose two novel contextual embedding-based approaches for Irony Detection and Emotion Recognition. We leverage social media-specific pre-training in the form of BERTweet - A language model pre-trained on English Tweets, along with either a Convolutional Neural Network or a Transformer Encoder. We empirically show that the addition of Convolutional Neural Networks or a Transformer Encoder results in improved performance when compared to a vanilla BERTweet model. Furthermore, we also compare CNNs and the Transformer Encoder as feature extractors, assessing the trade-off between the number of learnable parameters and performance. Finally, we also investigate the impact of partial and complete fine-tuning and analyze the trade-off between computational power and accuracy in the process. Experimental results demonstrate that our proposed methods achieve state-of-the-art performance on two standard benchmark datasets.**

***Index Terms*—Self-attention, Convolutional Neural Networks, Sentiment Analysis**

I. Introduction

Twitter[1] is one of the most widely used microblogging social media platforms that allows users to communicate real-time information about a variety of subjects and events. The platform's vast volumes of microblogging data may be utilised for a variety of essential analytic tasks [1], one of which is sentiment analysis. Over time sentiment analysis has gained prominence as a powerful field of research with a wide range of industrial applications. Individuals, businesses, and governments all benefit from it since they are interested in understanding people's views and opinions on various products and policies. As a result, Emotion Recognition on social media has become more important than ever. The identification of emotions in texts has multiple benefits in different areas, such as psychology [2], e-learning [3] and business intelligence [4]. Other than that, the growth of the social web has also promoted the use of metaphorical and creative language in public [5], including irony. Automatic Irony Detection has a lot of potential in the field of text mining, particularly for applications that need semantic analysis, such as author profiling or identifying online abuse.

Traditional Machine Learning-based approaches for Social media-specific Sentiment Analysis have often relied on the utilization of Support Vector Machine (SVM) [6] and Conditional Random Field (CRF) [44]. These methods, while effective, require manual feature engineering [7]. Deep Learning-based mechanisms like LSTMs and CNNs have achieved improved results on a number of Sentiment Analysis sub-tasks. However they have often done so by relying on recurrence and pre-trained word-embedding ensembles [8]–[10].

More recently, self-attention based models like Transformers [11] and BERT [12] have achieved state-of-the-art performance on several Natural Language Understanding (NLU) tasks. However, their success has largely covered common English domains such as Wikipedia, news and books. A point to note over here is that the nature of social media content is very different from these conventional sources, where the wording is more formal and traditional. Social media content is restrictive in length and often contains informal grammar, abbreviations and emoticons [13], [14]. Thus, it is not suitable to use Language Models that have been pre-trained on more conventional text-corpora as far as sentiment analysis on social media goes.

To this end, we propose two contextual embedding-based methods for Emotion Recognition and Irony Detection. We propose the use of BERTweet, based on [15], which is capable of generating contextual word embeddings utilizing the "multi-head self-attention" mechanism. The semantic level features are then extracted from the learned contextual embeddings using either a Convolutional Neural Network [16] or a Transformer Encoder [11]. We perform extensive analyses, experimenting with the number of trainable encoder layers in the BERTweet model. We also observe how its integration with CNNs and Transformer Encoder impacts the overall end-task performance. We assess our models on two standard benchmark datasets and outperform previous baselines by a significant margin.

Specifically, we make the following contributions through this paper:

- We propose two new contextual embedding-based ap-

§Equal Contributors
[1]https://twitter.com

2375-9259/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDMW53433.2021.00030

proaches for Emotion Recognition and Automatic Irony Detection based on Convolutional Neural Networks and the Transformer Encoder, thereby also leveraging BERTweet's pre-training on a large English Tweets Corpus.
- We achieve state-of-the-art results on two standard benchmark datasets, thereby also demonstrating that inclusion of a CNN or a Transformer Encoder further improves the performance when compared to a vanilla BERTweet model. Furthermore, we also compare our two proposed methods, experimentally analyzing the trade-off between the number of learnable parameters and performance.
- Lastly, we also experiment with layer-based ablations in the form of partial and complete fine-tuning and empirically show that increasing the number of trainable parameters beyond a certain point yields marginal returns, thereby also assessing the trade-off between computation power and accuracy.

The rest of the paper is structured as follows: Section 2 discusses previous approaches that have been used to tackle Emotion Recognition and Irony Detection, Section 3 describes the proposed methodology in detail, Section 4 describes the experimental setup, Section 5 discusses the results and analysis of our proposed methodology against previous methods and Section 6 concludes our paper.

II. Related Work

Sentiment analysis has long been an important focus of research. Due to the continuing growth of social networks, Irony Detection and Emotion Recognition, two of sentiment analysis' fundamental pillars, have grown increasingly significant. As a result, the scientific community has come up with a variety of solutions to these problems, using multiple variations of features and architectures.

Significant importance was given to features extracted from text which were then fed to machine learning algorithms. [21] made use of features such as "Imbalance" between the overall polarity of words and the rating of text, "Interjection" which indicated the occurrence of terms like "wow" and "huh" and "Emoticon" which indicated the occurrence of emojis. A combination of many such features was fed to Support Vector Machines (SVMs), Random Forest and Naive Bayes classifiers. Some solutions represented the sentence as a combination of pre-trained word embeddings and handcrafted features. [22] created a unified ensemble system and combined three different features generated using deep learning models and traditional methods. [23] developed an ensemble voting classifier with Logistic Regression and SVMs as component models. They combined pre-trained word and emoji embeddings with handcrafted features, including sentiment contrasts between various elements in a tweet. [45] presented a feature ensemble model which took into account lexical, word-type, semantic, position, and sentiment polarity of words.

While these approaches achieved excellent results, they depended on additional features retrieved from text, either solely or in combination. To solve this, several techniques relied entirely on the text and employed deep learning-based mechanisms to simulate sentence representations. [24] built an ensemble classifier of two deep learning models: a word-based and a character-based LSTM, both bidirectional in nature, to capture the semantic and syntactic information in tweets respectively. [25] proposed a Bi-LSTM architecture equipped with a multi-layer self-attention mechanism, pre-trained on a dataset from Semeval-2017. [26] explored the use of CNNs, RNNs and attentive RNNs and concluded that attentive RNNs performed the best while also demonstrating how the attention mechanism helps. [27], [29] demonstrated the combination of CNNs and LSTMs, whereas [28] illustrated the use of BiLSTMs, making use of character-level vector representations of words, based on [31]. [46] proposed a hybrid approach of CNNs and LSTMs which could handle diverse domains and analyse big social data. [47] presented an Attention-based Bidirectional CNN-RNN Deep Model (ABCDM) which extracted both past and future contexts of tweets by considering temporal information flow in both directions.

These approaches, however, have some drawbacks of their own. With approaches that utilise RNNs and LSTMs, it is impossible to avoid recurrence and the complexities that come with it. RNNs are often used to capture temporal relationships between words, however, they are strongly biased, i.e later words tend to be more dominant than the previous ones. Therefore, Transformers which eschew recurrence and make use of multi-head self-attention have achieved outstanding results on various NLP tasks [12], [41]–[43].

These architectures have also been put to use for Irony and Emotion Detection. [32] contextualized pre-trained Twitter word embeddings by employing the Transformer architecture. [33] proposed a pre-trained transformer-based network architecture which was further enhanced with the employment of a recurrent convolutional neural network. Some approaches went on to utilise word embeddings from pre-trained Transformers to enhance the results. [8] proposed an architecture that consisted of densely connected LSTMs based on pre-trained word embeddings, sentiment features using the AffectiveTweet package [34] and syntactic features. [9] presented Transformer based Deep Intelligent Contextual Embeddings (T-DICE) with attention-based BiLSTM. The contextual embeddings were a fusion of text representations - contextual [31], [35], traditional [36], part-of-speech, lexicon word and character representations. [37] proposed a network that combined the usage of word embeddings from XLNET, multi-channel CNNs and an attention mechanism that incorporated automatic weight adjustment to effectively improve the outcome of Emotion Recognition. [48] proposed a language representation model called SentiLARE, which introduced word-level linguistic knowledge including part-of-speech tags and sentiment polarity into pre-trained models.

Certain approaches utilise the principles of symbolic and sub-symbolic AI. For sentiment analysis, [49] used a combination of sub-symbolic and symbolic AI to automatically find conceptual primitives from text and related them to commonsense ideas and named entities. [50] used an ensemble of these concepts to combine top-down and bottom-up learning for polarity detection. [51] proposed a hybrid approach using Gated Convolutional Networks, aspect embeddings and the SenticNet framework for subtasks of aspect-based sentiment analysis. Certain approaches also made use of the stacked ensemble approach. [52] combined the outputs from a CNN, LSTM, BiLSTM and GRU using a stacking ensemble approach for sentiment analysis. [53] produced a stacking ensemble-based classifier model using LSTM, GRU, and Bi-GRU based base-level classifiers, as well as an LSTM based meta-level classifier, using three different word embeddings.

While there is a huge variety in approaches, we focus on the solutions that employ transformer-based models. Despite a huge number of these approaches being used to tackle Irony Detection and Emotion Recognition tasks, their pre-training data does not contain social media text, which includes emoticons, sarcasm, and colloquial terminology. To appropriately encode tweets, it is fitting to employ contextualised embeddings from models that adequately understand various characteristics of social media. As a result, we propose two new contextual embedding-based approaches for Automatic Irony Detection and Emotion Recognition based on Convolutional Neural Networks and the Transformer Encoder, specifically pre-trained on English tweets.

III. Methodology

In this section, we explain our proposed methodology in detail. The fundamental components of our model architecture are explained first, followed by the inference methodology.

A. Architecture Details

We present two contextual-embedding based approaches based on CNNs and the Transformer Encoder. Contextual embeddings are derived from BERTweet which are then fed as inputs to either a CNN or a Transformer Encoder.

a) Self-Attention and Transformer Encoder: A multi-headed self-attention mechanism and a position-wise fully connected feed-forward layer make up the Transformer Encoder, as illustrated in Fig. 1. A residual connection is employed between each of these two layers, and then layer normalisation is performed. Before the input embeddings are fed to the encoder, "positional encodings" are added to model the relative position of the tokens in the sequence.

The encoder employs the "Scaled Dot Product Attention" mechanism to encode each token in the sequence based on all the other relevant tokens. The following equation shows the formula to compute attention-focused weights:

$$Z = softmax(\frac{Q \times K^T}{\sqrt{(d_k)}})V \qquad (1)$$

where, $Q = (W_q E)$ is the Query vector, $K = (W_k E)$ is the Key vector, $V = (W_v E)$ is the Value vector. Here E is the embedding vector and W_q, W_k and W_v are trainable weight matrices.

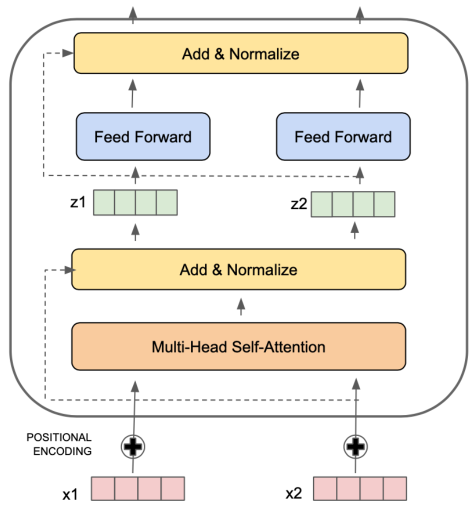

Fig. 1. Transformer Encoder

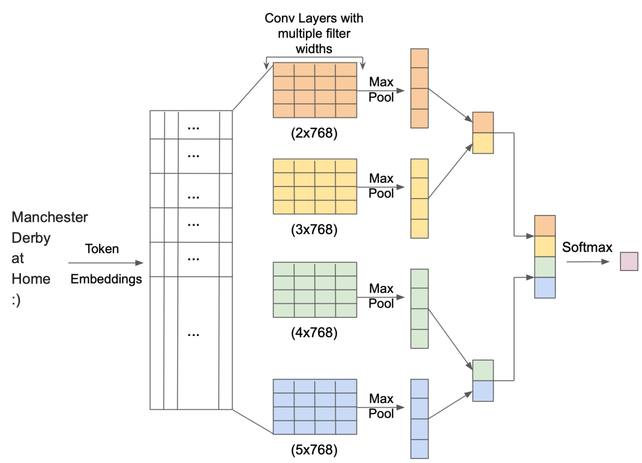

Fig. 2. CNN Model Architecture

The attention mechanism is further refined by employing "Multi Headed" attention. By generating multiple Z matrices for distinct sets of Key, Query, and Value weight matrices, the attention mechanism is utilized multiple times. All the output Z matrices are then concatenated into a single matrix and multiplied by an additional weight matrix, feeding the resultant output to the fully connected layer.

b) BERTweet: BERTweet [15] uses the same architecture as BERT$_{Base}$ [12], which in turn is significantly dominated by the Transformer Encoder, explained above. The pre-training procedure followed by BERTweet is inspired by RoBERTa [17], which re-establishes BERT's masked language modelling training objective to achieve better performance.

The model (BERTweet) was pre-trained with 80GB of data, which comprised ∼850M Tweets. Following a comprehensive pre-processing method, the Tweets were first normalised by

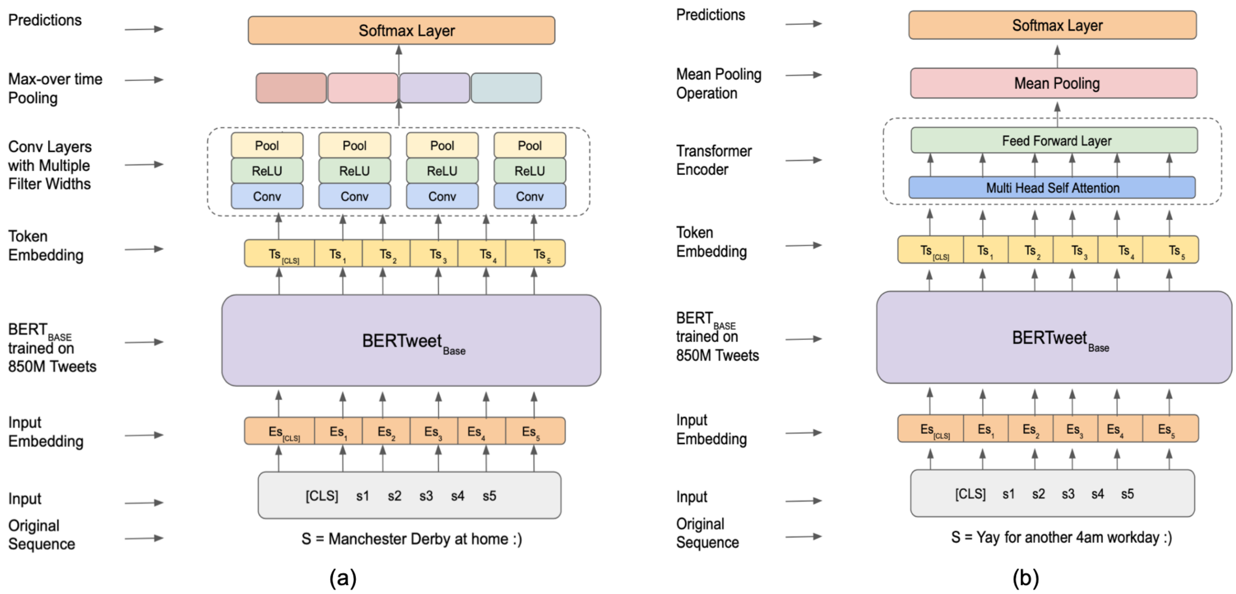

Fig. 3. Our Proposed Methodology - (a) Contextual Embedding based Convolutional Neural Network (CE-CNN). (b) Contextual Embedding based Transformer Encoder (CETE).

transforming user mentions and web/URL links to specific tokens like @USER and HTTPURL, respectively. Additionally, the emoji package was employed to convert emoticons to text strings.

We use the publicly available weights of the BERTweet model[2] which has **12** Encoder layers, **12** self-attention heads and an embedding size of **768**.

c) Convolutional Neural Networks: The CNN architecture we employ is shown in Fig. 2. Let $x_i \in R^{768}$ be the token embedding corresponding to the i^{th} token in the sequence. The resulting output representations from BERTweet (of length n) can be interpreted as:

$$x_{1:n} = x_1 \oplus x_2 \oplus ... \oplus x_n \tag{2}$$

where $\oplus$ is the concatenation operator.

The convolution operation applies a filter $w \in R^{m,768}$ to a m-word window to produce a new feature. A feature e_i is generated from a window of words $x_{i:i+m-1}$ by:

$$e_i = f(w \cdot x_{i:i+m-1} + b) \tag{3}$$

where b is the bias term and f represents the ReLU activation function. This filter is then applied to each possible window of words in the sequence $\{x_{1:m},\ x_{2:m+1},\ ...\ ,\ x_{n-m+1:n}\}$ to produce a feature map:

$$e = [e_1, e_2, ..., e_{n-m+1}] \tag{4}$$

[2]https://huggingface.co/vinai/bertweet-base

Following this, the feature map is subjected to max-over-time pooling, with the maximum value functioning as the feature corresponding to a certain filter.

A single feature is retrieved from a single filter using the above-mentioned process. To extract multiple features, the model employs several filters (of varying window sizes). The CNN configuration we employ consists of 400 parallel convolutional filters of four distinct sizes $(768 \times 2, 768 \times 3, 768 \times 4, 768 \times 5)$, each with 100 filters. To get the corresponding label, these features are concatenated and passed through a fully connected softmax layer. A dropout layer is adopted for regularization purposes.

B. Inference Methodology

As mentioned earlier, we combine BERTweet's pre-training with either a Convolutional Neural Network or a Transformer Encoder. In this subsection, we explain the two inference configurations in detail:

1) Contextual Embedding based Convolutional Neural Network (CE-CNN)
2) Contextual Embedding based Transformer Encoder (CETE)

In the CE-CNN approach, as shown in Fig. 3(a), our proposed approach works as follows: The input sequence is first tokenized and a token embedding is created for each token $s_i \in S$, where S is the input sequence. These embeddings, together with their accompanying attention masks, are then fed to the BERTweet model. The attention mask is a boolean vector that represents the tokens that should or should not be attended to by the model. Following this, multiple concurrent convolutional filters and max-pooling layers are employed to

TABLE I
PERFORMANCE COMPARISON (F1-SCORE) BETWEEN VANILLA BERT AND BERTWEET

Task	Vanilla BERT	BERTweet
Emotion Recognition	0.7527	0.8147
Irony Detection	0.4624	0.6206

extract multiple output features from BERTweet's resultant token embedding representations. The condensed vector obtained by concatenating these output features is then passed through a softmax layer to obtain the corresponding label.

In the case of the CETE approach, as illustrated in Fig. 3(b), the output representations from the BERTweet model are obtained in the same way as they were in the CE-CNN method. Following this, the resultant embeddings from the BERTweet model are fed to a Transformer Encoder. The encoder's output is mean-pooled to generate a condensed vector, which is finally passed through a softmax layer to get the corresponding label.

IV. EXPERIMENTAL SETUP

To evaluate the effectiveness of our approach, we carried out experiments on two standard benchmark datasets. In this section, we first present the description of the datasets, followed by the evaluation metrics we use. This is followed by the hyperparameter setup for all our experiments.

A. Dataset Description

a) SemEval-2018 Irony Detection: The dataset[3] was proposed by [18] and comprised two subtasks. We focus on subtask-B, which is a multi-class irony classification problem. The task's objective is to determine if a tweet is Non-Ironic, Verbally Ironic (through polarity contrast), Situationally Ironic, or contains any Other Form of Verbal Irony. The dataset comprises 3834 training instances and 744 test instances.

b) TweetEval Emotion Recognition: The dataset[4] was proposed by [19] and is a modification of the SemEval-2018 task proposed by [20]. The task is a multi-class emotion classification problem with the objective of recognising the emotion evoked by a tweet. The dataset comprises 3257 training instances, 374 validation instances and 1421 test instances and focuses on four major emotions: Anger, Joy, Sadness, and Optimism.

B. Evaluation Metrics

For the Irony Detection task, we evaluate ourselves against previous techniques on the basis of Accuracy, Precision, Recall and F1-Score. The official evaluation metric proposed by [19] for the Emotion Recognition task is the macro-averaged F1 Score. We use the same to ensure a fair comparison with previous benchmarks.

[3]https://github.com/Cyvhee/SemEval2018-Task3
[4]https://github.com/cardiffnlp/tweeteval/tree/main/datasets/emotion

C. Hyperparameter Setup

Both the CE-CNN and CETE methods use the same hyperparameter configuration. The input sequence's maximum length was set to **64**. We use PyTorch's Weighted Random Sampler to compensate for the imbalance in the datasets. For the CETE approach, the number of attention heads was set to **8** and the dropout value was chosen to be 5×10^{-1}. We use the Cross Entropy Loss function to calculate the loss. Adam was used as an optimizer, the batch size was chosen to be **8**, and the learning rate was set to 10^{-5}. The models were trained for **12** epochs on the Nvidia TESLA K80 GPU. All hyperparameters were chosen keeping the computational constraints in mind.

V. RESULTS

We compare ourselves to the baselines for Irony Detection and Emotion Recognition tasks in Table III and Table IV respectively.

We conduct experiments to determine how the number of tunable encoders affects the model's overall performance on both tasks. Experimental results show that increasing the number of tunable encoders improves performance, but not enough to compensate for the massive increase in the training cost. We also carry out experiments to analyze the influence of CNN and Transformer Encoder on the model architecture and the overall performance. When compared to Mean Pooling, incorporating these two architectures in the design yields considerably superior performance. Furthermore, we compare the CNN and the Transformer Encoder and discover that, while the latter produces better results, it does so at the expense of a significant increase in trainable parameters. We also analyse the influence of BERTweet by comparing it to the vanilla BERT architecture and observe a vast difference in results. All these results are tabulated in Table I and II.

Our main observations after performing this study have been:

1) Despite the varied nature of both tasks, the architectures we propose: CE-CNN and CETE, appear to generalise

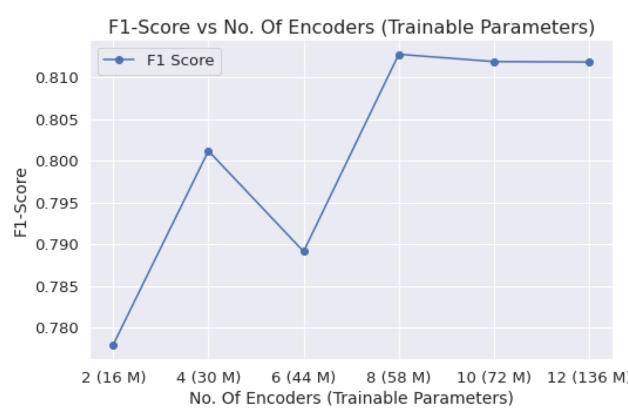

Fig. 4. Comparison between No. Of Encoders and F1-Score for CE-CNN approach

TABLE II
PERFORMANCE COMPARISON (F1-SCORE) IN THE TWO SETUPS - CE-CNN AND CETE. TR STANDS FOR TRAINABLE, MP STANDS FOR MEAN POOLING AND NO. OF TRAINABLE PARAMETERS (PARAMS) IS IN MILLIONS.

Method	No. of Tr. Encoders (No. of Tr. Params)	Emotion Recognition	Irony Detection
BERTweet + MP	4 (29 M)	0.7781	0.5698
	8 (57 M)	0.7878	0.5862
	12 (135 M)	0.7909	0.5825
CE-CNN	4 (30 M)	0.8012	0.5569
	8 (58 M)	**0.8128**	0.5999
	12 (136 M)	0.8119	**0.6111**
CETE	4 (35 M)	0.7665	0.5764
	8 (63 M)	0.8007	0.5908
	12 (141 M)	**0.8147**	**0.6206**

TABLE III
PERFORMANCE COMPARISON FOR IRONY DETECTION, RANKED BY F1-SCORE. ENC STANDS FOR NO OF TRAINABLE ENCODERS

Model	Acc	Precision	Recall	F1-Score
NIHRIO [38]	0.659	0.545	0.448	0.444
NLPRL-IITBHU [39]	0.603	0.466	0.506	0.474
THU_NGN [8]	0.605	0.486	0.541	0.495
NTUA-SLP [24]	0.652	0.496	0.512	0.496
UCDCC [40]	0.732	0.577	0.504	0.507
CETE (8 Enc.)	0.754	0.613	0.626	**0.591**
CE-CNN (8 Enc.)	0.784	0.667	0.613	**0.599**
CE-CNN (12 Enc.)	0.772	0.623	0.647	**0.611**
CETE (12 Enc.)	0.772	0.612	0.676	**0.621**

TABLE IV
PERFORMANCE COMPARISON FOR EMOTION RECOGNITION. ENC STANDS FOR NO OF TRAINABLE ENCODERS

Model	F1-Score
FastText [19]	0.652
RoBERTa-Twitter [19]	0.720
RoBERTa-Base [19]	0.761
RoBERTa-Retrained [19]	0.785
BERTweet [15]	0.793
CETE (8 Enc.)	**0.801**
CE-CNN (12 Enc.)	**0.812**
CE-CNN (8 Enc.)	**0.813**
CETE (12 Enc.)	**0.815**

quite well on both, Emotion Recognition and Irony Detection. This can be attributed to the pre-training of BERTweet on 850M English Tweets, as proven by the substantial performance difference between vanilla BERT[5] and BERTweet shown in Table I. (Both approaches use Mean Pooling.)

2) Experimenting with the number of encoder layers suggests that as the number of encoder layers increases, there is a relative increase in performance as well. This improvement, however comes at the cost of a significant increase in the trainable parameters and yields marginal returns after a certain point, as illustrated in Fig. 4. For instance, in the case of the Emotion Recognition task, while training the CE-CNN in its entirety ensures considerably better results, the same setup with four trainable encoders almost matches the same results while having ∼4x lesser trainable parameters.
3) In comparison to the mean pooling technique, CNNs and Transformer Encoders prove to be better semantic feature extractors. Experimental results suggest that both, CE-CNN and CETE despite having a smaller number of trainable encoder layers, outperform a fully fine-tuned BERTweet model on both tasks. This illustrates how the Transformer Encoder and CNN both aid BERTweet in capturing more semantic relationships, resulting in improved performance.
4) Experimental results suggest that when compared to CNNs, the Transformer Encoder achieves either equivalent or slightly better performance. While both techniques yield impressive results, the use of the Transformer Encoder (5 M) comes at the cost of ∼5x more trainable parameters when compared to a CNN (1 M), introducing a significant computational overload.

[5]https://huggingface.co/bert-base-uncased

VI. CONCLUSION

In this paper, we proposed two new contextual-embedding based approaches for Automatic Irony Detection and Emotion Recognition based on Convolutional Neural Networks and the Transformer Encoder. We used BERTweet to generate social media-specific contextual embeddings. Following this, we made use of either a Convolutional Neural Network or a Transformer Encoder to extract the semantic level features from the trained contextual embeddings. Through our experiments on the two benchmark datasets, we observed that both our approaches achieve state-of-the-art results on the two tasks. Furthermore, we also discovered that, while the Transformer Encoder has a slightly better feature extraction capability as compared to CNNs, it comes at the cost of ∼5x more trainable parameters. We also carried out partial and complete fine-tuning experiments, and empirically demonstrated that increasing the number of trainable parameters beyond a certain point yields marginal returns. The future direction of this work could include investigating the performance of the proposed methodology on multilingual social media data.

REFERENCES

[1] Ghani, N. A., Hamid, S., Hashem, I. A. T., & Ahmed, E. (2019). Social media big data analytics: A survey. Computers in Human Behavior, 101, 417-428.

[2] Cherry, C., Mohammad, S. M., & de Bruijn, B. (2012). Binary classifiers and latent sequence models for emotion detection in suicide notes. Biomedical informatics insights, 5(Suppl. 1), 147–154. https://doi.org/10.4137/BII.S8933

[3] Suero Montero, C., & Suhonen, J. (2014, November). Emotion analysis meets learning analytics: online learner profiling beyond numerical data. In Proceedings of the 14th Koli calling international conference on computing education research (pp. 165-169).

[4] E. Cambria, "Affective Computing and Sentiment Analysis," in IEEE Intelligent Systems, vol. 31, no. 2, pp. 102-107, Mar.-Apr. 2016, doi: 10.1109/MIS.2016.31.

[5] Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., & Reyes, A. (2015, June). Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015) (pp. 470-478).

[6] Joshi, A., Bhattacharyya, P.,& Carman, M. J. (2017). Automatic sarcasm detection: A survey. ACM Computing Surveys (CSUR), 50(5), 1-22.

[7] Barbieri, F., Saggion, H., & Ronzano, F. (2014, June). Modelling sarcasm in twitter, a novel approach. In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 50-58).

[8] Wu, C., Wu, F., Wu, S., Liu, J., Yuan, Z., & Huang, Y. (2018, June). Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 51-56).

[9] Naseem, U., Razzak, I., Eklund, P., & Musial, K. (2020, July). Towards improved deep contextual embedding for the identification of irony and sarcasm. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.

[10] Baziotis, C., Pelekis, N., & Doulkeridis, C. (2017, August). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017) (pp. 747-754).

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[12] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[13] Eisenstein, J. (2013, June). What to do about bad language on the internet. In Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies (pp. 359-369).

[14] Han, B., Cook, P., & Baldwin, T. (2013). Lexical normalization for social media text. ACM Transactions on Intelligent Systems and Technology (TIST), 4(1), 1-27.

[15] Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. arXiv preprint arXiv:2005.10200.

[16] Kim, Yoon. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 10.3115/v1/D14-1181.

[17] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[18] Van Hee, C., Lefever, E., & Hoste, V. (2018, June). Semeval-2018 task 3: Irony detection in english tweets. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 39-50).

[19] Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). Tweeteval: Unified benchmark and comparative evaluation for tweet classification. arXiv preprint arXiv:2010.12421.

[20] Mohammad, S., Bravo-Marquez, F., Salameh, M., & Kiritchenko, S. (2018, June). Semeval-2018 task 1: Affect in tweets. In Proceedings of the 12th international workshop on semantic evaluation (pp. 1-17).

[21] Buschmeier, K., Cimiano, P., & Klinger, R. (2014, June). An impact analysis of features in a classification approach to irony detection in product reviews. In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 42-49).

[22] Meisheri, H., & Dey, L. (2018, June). Tcs research at semeval-2018 task 1: Learning robust representations using multi-attention architecture. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 291-299).

[23] Rohanian, O., Taslimipoor, S., Evans, R., & Mitkov, R. (2018, June). Wlv at semeval-2018 task 3: Dissecting tweets in search of irony. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 553-559).

[24] Baziotis, C., Athanasiou, N., Papalampidi, P., Kolovou, A., Paraskevopoulos, G., Ellinas, N., & Potamianos, A. (2018). Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns. arXiv preprint arXiv:1804.06659.

[25] Baziotis, C., Athanasiou, N., Chronopoulou, A., Kolovou, A., Paraskevopoulos, G., Ellinas, N., ... & Potamianos, A. (2018). Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. arXiv preprint arXiv:1804.06658.

[26] Huang, Y. H., Huang, H. H., & Chen, H. H. (2017, April). Irony detection with attentive recurrent neural networks. In European Conference on Information Retrieval (pp. 534-540). Springer, Cham.

[27] González, J. Á., Hurtado, L. F., & Pla, F. (2018, June). ELiRF-UPV at SemEval-2018 Tasks 1 and 3: Affect and irony detection in tweets. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 565-569).

[28] Ilić, S., Marrese-Taylor, E., Balazs, J. A., & Matsuo, Y. (2018). Deep contextualized word representations for detecting sarcasm and irony. arXiv preprint arXiv:1809.09795.

[29] Ortega-Bueno, R., & Pagola, J. E. M. (2018). UO IRO: Linguistic informed deep-learning model for irony detection. EVALITA Evaluation of NLP and Speech Tools for Italian, 12, 166.

[30] Ratadiya, Pratik, and Deepak Mishra. "An attention ensemble based approach for multilabel profanity detection." In 2019 International Conference on Data Mining Workshops (ICDMW), pp. 544-550. IEEE, 2019.

[31] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[32] González, J. Á., Hurtado, L. F., & Pla, F. (2020). Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. Information Processing & Management, 57(4), 102262.

[33] Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications, 32(23), 17309-17320.

[34] Bravo-Marquez, F., Frank, E., Mohammad, S. M., & Pfahringer, B. (2016, October). Determining word-emotion associations from tweets by multi-label classification. In 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI) (pp. 536-539). IEEE.

[35] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

[36] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[37] Yang, C. T., & Chen, Y. L. (2020, June). DACNN: Dynamic Weighted Attention with Multi-channel Convolutional Neural Network for Emotion Recognition. In 2020 21st IEEE International Conference on Mobile Data Management (MDM) (pp. 316-321). IEEE.

[38] Vu, T., Nguyen, D. Q., Vu, X. S., Nguyen, D. Q., Catt, M., & Trenell, M. (2018). Nihrio at semeval-2018 task 3: A simple and accurate neural network model for irony detection in twitter. arXiv preprint arXiv:1804.00520.

[39] Rangwani, H., Kulshreshtha, D., & Singh, A. K. (2018, June). Nlprl-iitbhu at semeval-2018 task 3: Combining linguistic features and emoji pre-trained cnn for irony detection in tweets. In Proceedings of the 12th international workshop on semantic evaluation (pp. 638-642).

[40] Ghosh, A., & Veale, T. (2018, June). Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 570-575).

[41] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[42] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

[43] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[44] Patra, B. G., Soumik, M., Das, D., & Sivaji, B. (2014). Ju_cse: A conditional random field (crf) based approach to aspect based sentiment analysis.
[45] Phan, H. T., Tran, V. C., Nguyen, N. T., & Hwang, D. (2020). Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model. IEEE Access, 8, 14630-14641.
[46] Behera, R. K., Jena, M., Rath, S. K., & Misra, S. (2021). Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. Information Processing & Management, 58(1), 102435.
[47] Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems, 115, 279-294.
[48] Ke, P., Ji, H., Liu, S., Zhu, X., & Huang, M. (2020, November). Sentilare: Linguistic knowledge enhanced language representation for sentiment analysis. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6975-6988).
[49] Cambria, E., Poria, S., Hazarika, D., & Kwok, K. (2018, April). SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).
[50] Cambria, E., Li, Y., Xing, F. Z., Poria, S., & Kwok, K. (2020, October). SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In Proceedings of the 29th ACM international conference on information & knowledge management (pp. 105-114).
[51] Chandra, J. K., Cambria, E., & Nanetti, A. (2020, November). One belt, one road, one sentiment? a hybrid approach to gauging public opinions on the new silk road initiative. In 2020 International Conference on Data Mining Workshops (ICDMW) (pp. 7-14). IEEE.
[52] Mohammadi, A., & Shaverizade, A. (2021). Ensemble deep learning for aspect-based sentiment analysis. International Journal of Nonlinear Analysis and Applications, 12, 29-38.
[53] Subba, B., & Kumari, S. (2021). A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. Computational Intelligence.