BERT-Based Emotion Recognition

Team Gold

Aiman Haider **Maobin Guo** **Pranav Manjunath** **Xinyi Pan**

Abstract

Multi-class Emotion Detection in Text has been a conventionally difficult task given the presence of subjectivity, subtleties and stylizations used in text. While a number of conventional and deep learning techniques have been used, the report finds the BERT Model to be the best performing so far, and develops an application to classify text from an external dataset into the seven basic emotion classes.

1 Introduction

Understanding another person's emotions is critical in interpersonal communication. Discussion on emotions has a long history and has become more popular in contemporary business scenarios. Since Greek and Roman times (Poria et al., 2017), decision-makers utilized public opinion to advocate their policies and formed democracies. Nowadays, public opinion and emotions are more significant not only in politics but also valuable for product reviews and promotion (Cambria, 2016). The proliferation of online activities, the exponential growth of product and service reviews on transparent platforms and opinions on social media endow industry and academia with affective computing (Poria et al., 2017), which is the capacity for recognizing, inferring, and deciphering human emotion. Affective computing serves an increasingly central role in spam detection technology, recommendation systems, and most importantly, business intelligence, where the corporate interest lies (Cambria, 2016).

Through this paper and project, we aim to look at the domain of affective computing by performing multi-class classification of emotions by comparing different machine learning algorithms, ranging from logistic regression, LSTM, to BERT. The dataset used is the International Survey on Emotion Antecedents and Reactions (ISEAR), which contains 7,666 records with 7 emotion labels. In this paper, we first review relevant literature in Section 2. The methodologies adopted are then explained in Section 3, and the results are discussed in Section 4. We also developed an interactive application on unlabeled Facebook Comments with Python, which is shown in Section 5.

2 Background

Affective computing which deals with identification, processing, interpretation, and simulation of human emotional states intelligently from data (Poria et al., 2017), consists of sentiment analysis and emotion recognition. Though there are close similarities between sentiments and emotions, the subtleties between these two subjective terms also exist (Munezero et al., 2014). Sentiments are more stable and targeted while emotions are short-lived and transient. Accordingly, the mining and analysis of sentiments and emotions are semantically distinct from each other.

In addition, sentiment analysis and emotion recognition have different levels of granularity (Poria et al., 2017). The former is characterized as coarse-grained affect recognition since it performs a binary classification task in general with outputs such as positive versus negative (Poria et al., 2017), (Cambria, 2016). In contrast, the latter is referred to as fine-grained affect recognition (Poria et al., 2017), (Cambria, 2016) as it is based on a larger set of emotion labels including happiness, sadness, anger, and surprise (Ekman, 1992).

In light of the similarity between detecting coarse-grained and fine-grained emotions, approaches to emotion recognition can be mainly categorized into lexicon-based techniques and statistical methods (Cambria, 2016). The lexicon-based approach, also commonly referred to as a knowledge-based approach (Cambria, 2016), exploits the lexical, morphological, and syntactic characteristics of texts to perform text-based emotion recognition. Extensive literature has proposed, evaluated, and leveraged lexicon-based techniques such as WordNet-Affect (Strapparava and Valitutti, 2004), SentiWordNet (Esuli and Sebastiani, 2006), and SenticNet (Cambria et al., 2014) in the last two decades. Nonetheless, Cambria (Cambria, 2016) states that the lexicon-based approach is weak when the mining and detection of emotions are involved with linguistic rules. Moreover, emotion recognition with knowledge-based techniques highly relies on human-generated baselines, such as a comprehensive corpus annotated by humans. However, given that the original intention of humans to explore this approach is to enable machines to detect emotions automatically, knowledge-based techniques are becoming obsolete, and therefore scholars have proposed and increasingly adopted another category of methodologies viz. statistical methods (Poria et al., 2017), (Cambria, 2016).

Among these, there has been a special focus on supervised machine learning techniques (Poria et al., 2017), including Naive Bayes, Maximum Entropy, and Support Vector Machines (Pang et al., 2002). Deep learning, an unsupervised machine learning technique, is also widely exploited in the existing literature of emotion recognition (Poria et al., 2017). One of the advantages of statistical methods is that it provides more accurate and reasonable classification compared to knowledge-based approaches. For instance, Pang et al. (Pang et al., 2002) found that supervised machine learning techniques significantly outperform knowledge-based techniques in terms of classification accuracy. Also, statistical approaches have been found more powerful than other alternatives in predicting lexical affinity between words and the frequency of word co-occurrence (Cambria, 2016). This report, too, adopts the deep learning approach to the task of emotional classification.

3 Methodology

A few techniques are explored first to understand their performances and appropriateness, and then an application is developed using the BERT model, which showed the best results.

The dataset used is a dataset collected by the International Survey on Emotion Antecedents and Reactions (ISEAR) during the 1990s. For building the ISEAR, 1,096 participants who have different cultural backgrounds completed questionnaires about experiences and reactions for seven emotions including anger, disgust, fear, joy, sadness, shame, and guilt, 7 major emotion classes on which our study is based [1]. This dataset consists of 7,666 sentences from around 3000 respondents in 37 countries on all 5 continents. Extensive literature has evaluated this dataset for emotion recognition and achieved impressive performance. Therefore, we decided to utilize this dataset for our classification purpose. The dataset is split into 75% for training and 25% for testing.

3.1 Multinomial Logistic Regression

Given the task of multinomial classification of emotions, the first classification method we applied is multinomial logistic regression. Multinomial logistic regression, also called softmax regression, is a supervised machine learning algorithm for the classification of more than two classes (Jurafsky and Martin, 2009). In multinomial logistic regression, the dependent variable is categorical. The emotions are classified by evaluating the probability of this dependent variable to belong to each potential emotional class. During this process, one emotion class is chosen as the baseline category. Our multinomial logistic model is therefore a set of logistic regression models for each emotional class, compared to the baseline.

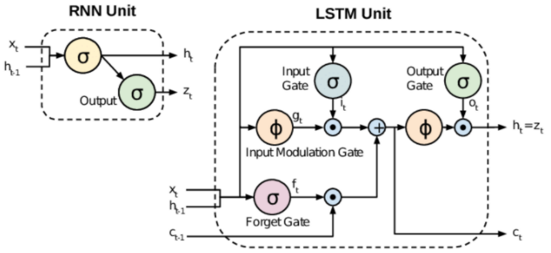
We built a baseline by applying the multinomial logistic regression to TF-IDF representations of emotions. In this experiment, the dataset was first split into the training set and testing set, of which the size is 30% of the original dataset. Next, the TF-IDF vectorizer was employed to transform training and testing sets with texts. Finally, we applied the multinomial logistic regression to the training set and tuned hyperparameters of the model to predict. The overall accuracy was 55% across all emotion labels.

3.2 Long Short Term Memory and Bidirectional Long Short Term Memory

Next, we attempt two popular methods in Deep Learning - LSTM and BiLSTM, which stands for Long Short Term Memory and Bidirectional Long Short Term Memory. The LSTM is a special type of Recurrent Neural Network (RNN), which

[1] https://www.unige.ch/cisa/research/materials-and-online-research/research-material/

Figure 1: Structure of BiLSTM



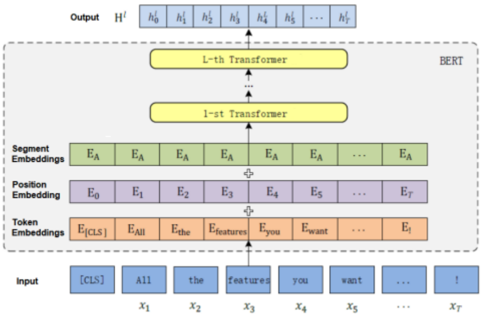
"remembers" and uses immediate previous values through feedback loops, that can learn long term patterns based on a combination of immediate and some old retained memory as can be seen through its architecture as in the figure 1.

First, we try using LSTM with GLove Embeddings on the dataset. The text is preprocessed by casing, tokenization, and lemmatization. The tokens are then vectorized using the dictionary entered by the pre-trained vector model which gives us the corresponding GloVe Embedding Vectors (Pennington et al., 2014). Next, the vectorized text is encoded with LSTM and then passed through a softmax layer for classification. Next, we try using the BiLSTM, which is a variant of LSTM that processes the information in dual ways across the starting and terminating neurons of the network for faster training and better flow of information. It is a sequence processing model that consists of two LSTMs where one takes the input in a forward direction, and the other in the backward direction. Due to this, BiLSTM increases the amount of information available to the network, improving the context available to the algorithm. Unsurprisingly, it gives better results.

3.3 Bidirectional Encoder Representations from Transformers

We then turn to BERT, short for Bidirectional Encoder Representations from Transformers. BERT is a state-of-the-art pre-training language representation. It is based on the transformer framework where the input text sequence is deeply represented by encoding and decoding (Devlin et al., 2018) and uses a multi-head self-attention mechanism (Li et al., 2019). The attention mechanism extracts the different "role" information in word representations, namely Q (Query), K (Key), and V (Value). The attention weight is obtained by scaling the dot product of the Q of the current word and the K of the word being noticed and then obtained by the softmax layer. The attention weight is assigned to the V of the current word and then added to its representation. BERT uses Masked LM (Masked Language Model) for training. These allow for a few words in the token sequence (sentence) to be masked and facilitate its prediction as also in predicting the next "sentence". The goal of Masked LM ensures that BERT can obtain a representation of the target word that combines left and right bidirectional context information.

Figure 2: Structure of BERT



BERT mainly functions to obtain a representation of a sentence (word sequence), which is expected to reflect the relationship between words. To this end, it has designed additional tags that indicate the beginning, end, and connection of the word sub-sequence; and the position information of the word is also added during the processing. The real words, symbols, marks, and segmented units, are referred to as tokens. As shown in 3.3, the pre-training input of BERT is embedded by token Entry (token Embeddings), segment embeddings (Segment Embeddings), and position embedding (Position Embeddings). These then go through the layers of encoders.

Let L be the number of BERT layers, *dimh* represents the dimension of the word, Transformerl represents the mapping of 1 layer $(l = 1, 2, \cdots, L)$ determined by Transformer, $x = (x_1, x_2, \cdots, x_T)$ is a text input to BERT, and $H^l = \left(h_1^l, h_2^l, \cdots, h_T^l\right)$ are recorded as the output of the l^{th} layer. Hence, $H_l = Transformerl\,(H_l - 1)$. Finally, the output of BERT is $H^L = \left(h_1^L, h_2^L, \cdots, h_T^]\right)$.

Google provides two pre-trained versions of the BERT model: BERT_BASE (L=12, H=768, A=12, Total Parameters=110M) and BERT_LARGE (L=24, H=1024, A=16, Total Parameters=340M) where L represents the number of layers (Transformer Blocks), H represents the space dimension, A represents the number of attention heads, and Total Parameters represents the number of parameters. We use the BERT_BASE version in this report and find it to be the best performing so far. We also compared this to the baseline results for the data and find that it outperformed those results.

4 Results

4.1 Results using the BERT Model

The results for the BERT model on the dataset can be seen in Table. As is clearly evident, the BERT model performs much better in comparison to all other models mentioned above. The Attention mechanism is a major reason for this as it does not distribute the weightage of each embedding equally rather focuses on some.

Emotion	Precision	Recall	F1-Score
Disgust	0.70	0.69	0.70
Fear	0.77	0.77	0.77
Shame	0.59	0.51	0.55
Guilt	0.62	0.64	0.63
Anger	0.60	0.59	0.59
Sadness	0.71	0.73	0.72
Joy	0.81	0.90	0.85
Accuracy		0.69	
Macro avg	0.69	0.69	0.69
Weighted avg	0.69	0.69	0.69

Table 1: Output of BERT

It reports an overall accuracy of 69% with a macro F1, Recall and Precision of 0.69. This model, thus, also seems to indicate a good balance of sensitivity and specificity overall. Also, it should be pointed out that its results have been pretty close to that best metrics reported for the dataset which is 67%. [2]

Further, it can be seen that the model has a much better performance when it comes to predicting joy than other emotions. It can be seen that shame, anger and the upsetting emotions do not fare as well, in comparison to joy related sentences. This can possibly be because of the preponderance of so many like upsetting emotions where it might be difficult for the model to get over the subtleties. Besides, the limitations due to linguistic subjectivity and stylizations are other possible reasons. Nonetheless, it performs well on these parameters when compared to other models. This is quite visible through the confusion matrix, too.

5 Application

To practically apply this study in the real world, we built a Python application using the Tkinter library in Python and ran our model on unlabeled Facebook Comments from Kaggle. The application fetches random comments from Kaggle, sends it through the trained BERT model using the ISEAR dataset and displays the predicted emotion along with the comment. The application uses the sample Panda images found on Powtoon and associates each emotion to a respective Panda Emoji. Figures in the appendix show three sample screenshots from the application.

6 Conclusions

7 Acknowledgement

References

[Cambria et al.2014] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. 2014. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. volume 2, 07.

[Cambria2016] E. Cambria. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107.

[Devlin et al.2018] 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*.

[Ekman1992] Paul Ekman. 1992. Facial expressions of emotion: New findings, new questions. *Psychological Science*, 3(1):34–38.

[Esuli and Sebastiani2006] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May. European Language Resources Association (ELRA).

[Jurafsky and Martin2009] Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

[Li et al.2019] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*.

[2] `https://arxiv.org/pdf/1908.01587.pdf`

[Munezero et al.2014] M. Munezero, C. S. Montero, E. Sutinen, and J. Pajunen. 2014. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE Transactions on Affective Computing*, 5(2):101–111.

[Pang et al.2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics, July.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

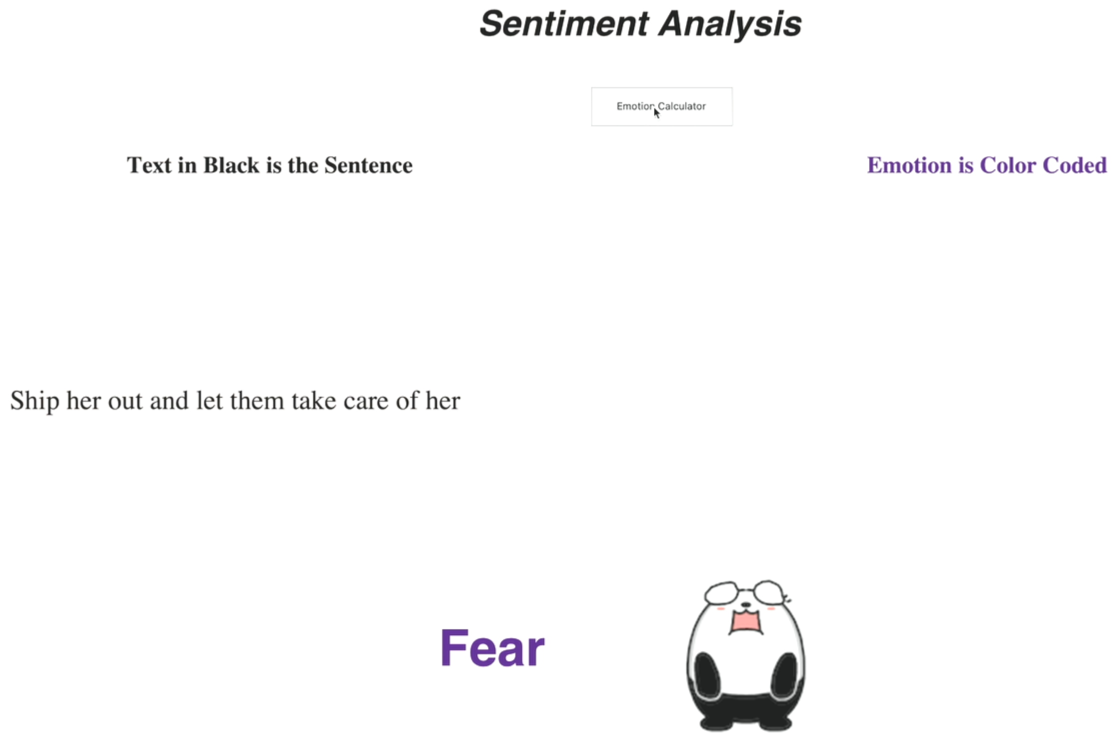
[Poria et al.2017] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. 2017. A review of affective computing. *Inf. Fusion*, 37(C):98–125, September.

[Strapparava and Valitutti2004] Carlo Strapparava and Alessandro Valitutti. 2004. WordNet affect: an affective extension of WordNet. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May. European Language Resources Association (ELRA).

Appendix

Figure 3 shows that the model labels the comment “This was from 2006. We have come a long way since then. In the wrong direction” as Fear.

Figure 3: Example of Fear Output



Sentiment Analysis

Text in Black is the Sentence

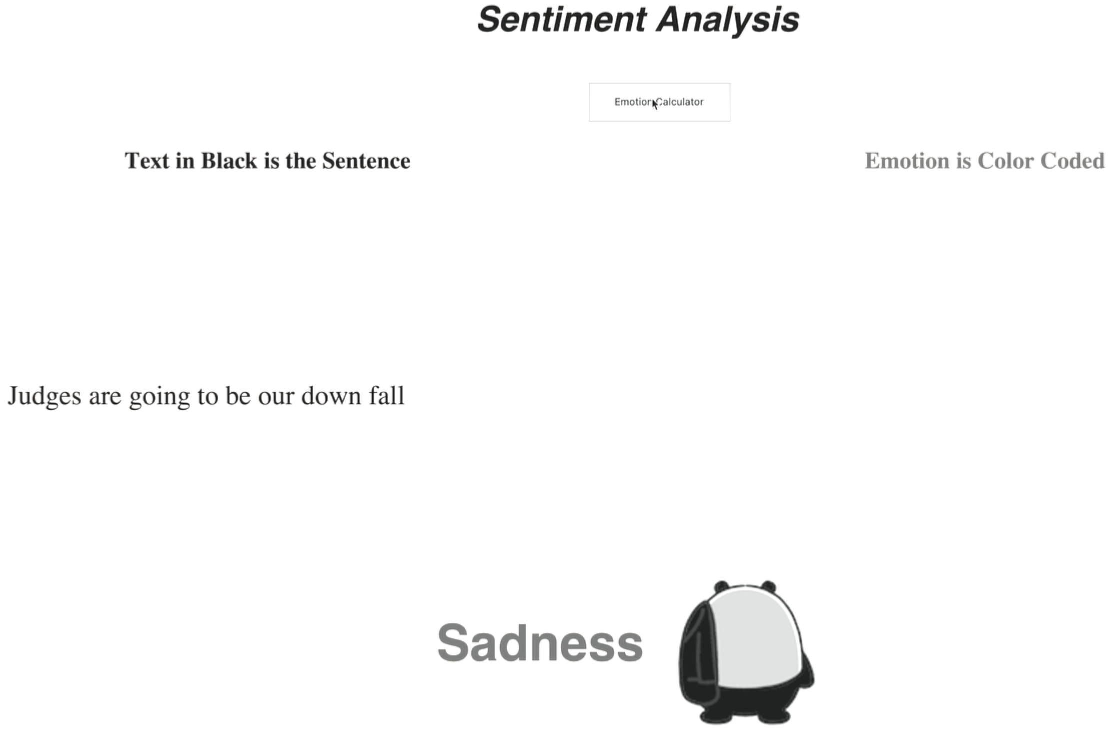
Emotion is Color Coded

Ship her out and let them take care of her

Fear

Figure 4 illustrates that the model labeled the comment “I’d like to stop it too. . . but crooks be crooks” as Sadness.

Figure 4: Example of Sadness Output



Sentiment Analysis

Text in Black is the Sentence

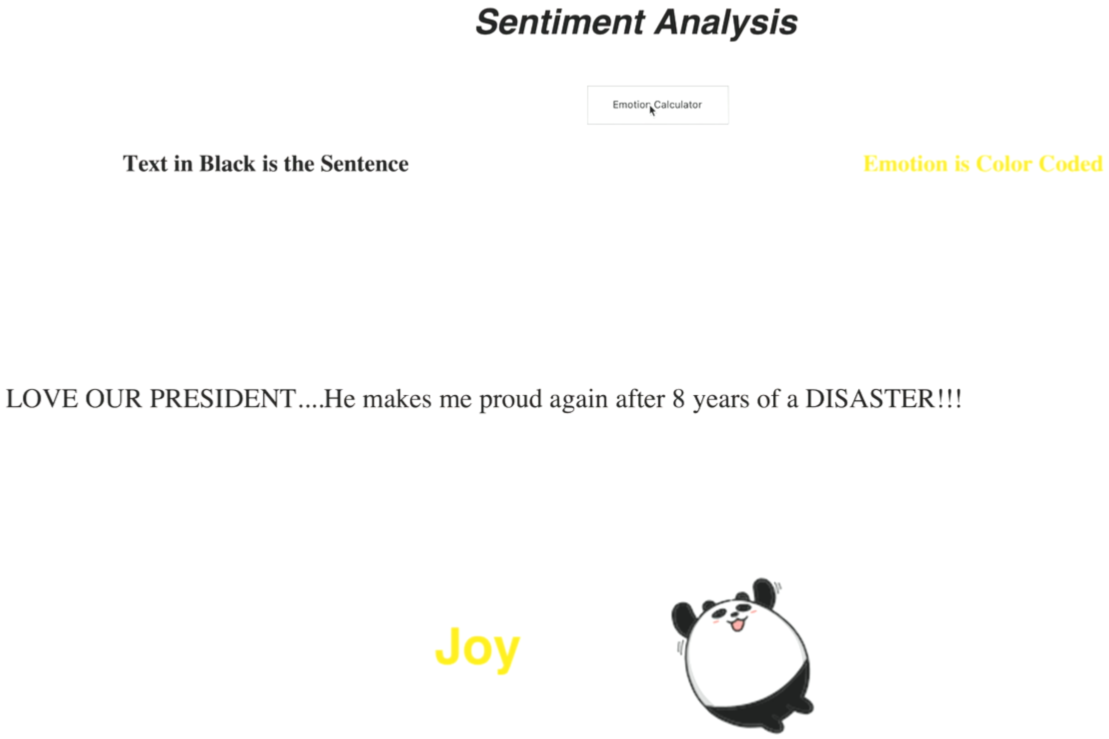
Emotion is Color Coded

Judges are going to be our down fall

Sadness

Figure 5 depicts that the model labeled the comment “LOVE OUR PRESIDENT. . . .He makes me proud again after 8 years of a DISASTER!!!” as Joy.

Figure 5: Example of Joy Output



Sentiment Analysis

Text in Black is the Sentence

Emotion is Color Coded

LOVE OUR PRESIDENT....He makes me proud again after 8 years of a DISASTER!!!

Joy