Task 1: Detecting Implicit Emotions

Arun Rajendran[1] *Ife Adebara*[2] *Una Chow*[2]
Department of Mathematics[1], *Department of Linguistics*[2]
arun95@math.ubc.ca, {*ife.adebara,una.show*}*@ubc.ca*
University of British Columbia

Abstract

In this paper, we explore various approaches for detecting implicit emotions from data and describe the best performing approach. We adopt deep learning methods including long short-term memory (LSTM), LSTM with attention (LSTM-A) and Convolutional Neural Network (LSTM-CNN). We also experiment with traditional machine learning methods like Linear Support Vector Machine (LinearSVM), Naive Bates (NB), Logistic Regression (LogReg) and Voting Classifier. Our LSTM-A model performs best for this task with a F1-Score of 0.6231.

1 Introduction

Sentiment analysis, which is also called opinion mining, is an area of research that focuses on analyzing the opinions, sentiments, attitudes, emotions and appraisals of people towards entities and attributes expressed in a written document (Liu, 2012; Mohammad and Turney, 2013). The entities may be opinions about services, political events or any issue that expresses subjective information. This implies that sentiment analysis focuses on subjective information rather than objective information like facts (Taboada, 2016).

Sentiment analysis is useful in almost every social domain because opinions have a strong influence on behaviour as well as on decisions in human endeavour (Feldman, 2013; Liu, 2012). How we perceive things, our reality, beliefs, and expectations are often hinged on the opinions of others. For instance, people will often view the opinions of others on several health related issues. Many patients often learn from others' experience about medication, different ailments, lifestyle changes resulting from illness, choosing a health care centre, among many other concerns related to a patient's health, and they use such knowledge to inform their own decisions. Patients often share this information wrapped in their own sentiments and emotions, and even though these reviewers are often strangers, a bad review about a medication for instance can prevent a patient from using that medication. Thus it is important to have a system that is able to pass judgment on subjective information.

In this study, we conduct an emotion classification experiment using Twitter data on five machine learning models: LinearSVM, NB, LSTM, LSTM-A, and LSTM-CNN. Tweets that are likely to express emotions are used. Each tweet contains an emotion word (i.e., a trigger word) that has been removed and replaced with the neutral tag "[TRIGGERWORD]". Our goal is to compare how well the models can correctly predict the conveyed emotion of the trigger word from the remaining text in the tweet.

The rest of the paper is organized into four sections. Section 2 provides literature of related work. Section 3 describes the Twitter data. Section 4 details our methods, including the models. Section 5 presents the results with analysis and discussion. Finally, section 6 concludes while section 7 provides direction for future work.

2 Related Work

Several approaches have been employed for sentiment analysis in recent years. Emoticons that correspond to affective states such as happy, sad, etc have been explored (Yang et al., 2007; Fraisse and Paroubek, 2014). Another approach used a compositional semantics approach to create an emotion lexicon from emotion labeled news articles. Crowdsourcing has also been used for determining the emotion in text (Mohammad and Turney, 2013; Staiano and Guerini, 2014). A crowdsourc-

ing approach was used to build an emotion lexicon from a thesaurus. Amazon Mechanical Turk workers were then asked to provide emotion information for those terms.

Abdul-Mageed and Ungar (2017) investigated the use of hashtags to generate emotion labels in order to eliminate costly manual labeling. Their gated recurrent neural network is the first to accurately detect 24 fine-grained types of emotion from Twitter datasets at an average of over 87 (Abdul-Mageed and Ungar, 2017).

Yang Liu and Xie Zhu(2018) expanded the original text data by mining related semantic information and used a ensemble classifier composed of SVM,kNN and HMM to analyze the emotion of short text of micro blog (Liu and Zhu, 2018). Cecilia et al (2005) worked on classifying the emotional affinity of sentences in the narrative domain of childrens fairy tales and showed naive bayes model with bag of words approach yields encouraging results (Alm et al., 2005).

Yoon Kim (2014) demonstrated the effectiveness of simple CNN model with pre-trained word vectors on sentiment analysis tasks (Kim, 2014). Xiang Zhang et al (2015) showcased that character level CNN model achieves state-of-the-art or competitive results on text classification datasets (Zhang et al., 2015). Ccero Nogueira dos Santos et al (2014) proposed a new deep CNN that exploits from character to sentence-level information to perform sentiment analysis of short texts (dos Santos and Gatti, 2014).

Yequan Wang et al (2016) explored an Attention-based Long Short-Term Memory Network for aspect-level sentiment classification and exhibited that it achieves state of the art results (Wang et al., 2016c). Yukun Ma et al (2018) put forth a novel solution to targeted aspect-based sentiment analysis by augmenting the long short-term memory (LSTM) network with a hierarchical attention mechanism consisting of a target level attention and a sentence-level attention (Ma et al., 2018).

Jin Wang et al (2016) proposed a regional CNN-LSTM model for doing dimensional sentiment analysis (Wang et al., 2016a). Wang et al (2016) combined CNN and RNN to take advantage of the coarse-grained local features generated by CNN and long-distance dependencies learned via RNN for sentiment analysis of short texts (Wang et al., 2016b).

Recently, the implicit emotion task has also been tackled using an ensemble of models like Bag of Words Logistics Regression Model, LSTM, Bidirectional LSTM, and more (Alhuzali et al.).

3 Dataset

We obtained sample English data from the WASSA Implicit Emotions Shared Task (IEST) 2018. The original datasets comprised 153,383 tweets for training, 9,591 tweets for validation, and 28,757 tweets for testing. Each tweet was paired with an emotion word label from the set anger, disgust, fear, joy, sad, surprise that was identical or semantically similar to the removed trigger word in the tweet. (Klinger et al., 2018).

4 Methods

4.1 Preprocessing

To proceed with this task, we had to preprocess the training and validation data. The files were in CSV format, delimited with commas and semicolons, and some lines contained special characters that were interpreted as new columns. Thus we used regular expressions to remove usernames, URLs, hash tags, and punctuations and emojis. The tweets were tokenized using SpaCy's (https://spacy.io/) load function option.

After cleaning the data, we had 151245 instances for the training data, and 9474 instances for the validation data.The labels for the development file were provided in a separate file and labels for test file were not provided. The fake labels in the validation file were replaced with original labels and validation set was used for hyperparameter tuning.

4.2 Traditional Machine learning Models

We used bag of words approach including a wide range of ngram values from 1 to 4 followed by Tf-Idf. Different range of values were tried for ngram and this seemed to perform better on validation set. We also tried removing stop words to see if it improves the F1 score but it reduced the F1 score and was not used in the final model. The target labels were encoded using label encoder. The macro F1 score was used as metric for measuring the performance of different models.

4.3 Deep learning Models

For the model's embedding layer, we converted the tokenized words to statistics using GloVe's

Common Crawl pretrained model, which contains 840 billion tokens and 2.2 million cased vocabularies, resulting in 300-dimensional vectors. GloVe creates global vectors for word representation based on frequencies of pairs of co-occurring words and stores some semantic information (Pennington et al., 2014). Since the trigger words are of the same part of speech (i.e., adjectives), the semantics rather than the syntactic relation of their contextual words are more likely to help to predict the emotion labels. The embedding weights in the deep learning models were fixed at pre-trained glove embedding weights. Due to computational constraints, all the deep learning models were run for only 5 epochs and the F1 score might be improved if they are run for more iterations. Also, the macro F1 score was calculated as an average of different validation batch F1 scores. We use a Adam optimizer with learning rate of 0.001 with Cross entropy loss for optimization.

4.3.1 CNN

The model consists of three filters followed by a dropout layer. The different outputs are concatenated and max pooling is applied on the result followed by dropout and a fully connected layer. We use a log softmax layer to obtain the probability of different classes. We also use a learning rate scheduler to decrease the learning rate by a factor of 0.1 after 2 iterations to reduce over-fitting.

Hyper-Parameter	Value
Embed Dimension	300
Kernel Size	[3,4,5]
Kernel Dim	256
Batch Size	64
Dropout	0.5

Table 1: Network Architecture & Hyper-Parameters of CNN model

4.3.2 LSTM

The model consists of lstm layer followed by a dropout layer. The results are passed through another fully connected layer with relu activation followed by a dropout layer and a fully connected layer. We also use a function that decreases the learning rate by a factor of 0.1 after observing an increase in validation loss for 1 iteration, in order to reduce overfitting.

4.3.3 LSTM with Attention

This model is a variant of LSTM model with lstm layer followed by self attention function that takes in final hidden state and output to give new hidden state using soft attention. The rest of the model architecture remains same as LSTM model. We also use a learning rate scheduler to decrease the learning rate by a factor of 0.5 after 2 iterations to reduce overfitting. The model was run for 1 epoch, then it was saved. Later, we loaded the model and ran it for 4 more epochs.

Hyper-Parameter	Value
Embed Dimension	300
Hidden Dim	256
Number Layer	1
FC1 Dim	128
Batch Size	64
Dropout	0.5

Table 2: Network Architecture & Hyper-Parameters of LSTM and LSTM with Attention models

4.3.4 LSTM + CNN

This model is variant of the LSTM model with attention, adding a CNN layer on top. The new hidden state from attention function passed to three sequential models of 1D convolution filters,batch normalization,relu activation function and a dropout layer. The results are concatenated and passed to series of layers in the order, dropout layer, fully connected layer, dropout layer and a fully connected layer. The results are passed through a log softmax layer to get probabilities of different classes.We also use a learning rate scheduler to decrease the learning rate by a factor of 0.5 after 2 iterations to reduce overfitting.

Hyper-Parameter	Value
Embed Dimension	300
Hidden Dim	256
Number Layer	1
Kernel Size	[1,3,5]
Kernel Dim	100
FC1 Dim	128
Batch Size	64
Dropout	0.5

Table 3: Network Architecture & Hyper-Parameters of LSTM+CNN model

5 Results

Table 1 shows the average macro F1 score of Traditional machine learning models across the validation set. Table 2 shows the average macro F1 score of deep learning models across the validation set. The scores obtained for the deep learning models were only run for 5 epochs and might result in an increase if the model is run for more number of epochs.

Task	Macro F1-Score
Naive Bayes + BOW	0.5852
LinSVM + BOW	0.5977
Log Reg + BOW	0.5724
NB + BOW + TF-IDF	0.5727
LinSVM + BOW + TF-IDF	0.6124
Log Reg + BOW + TF-IDF	0.5825
Ensemble	0.6046

Table 4: Traditional Machine Learning Models

Among traditional machine learning models, LinearSVM model performs better than Logistic regression model and Naive Bayes model. LinearSVM model with Bag of words and TF-IDF transformation performs the best among all the models. The Ensemble model performs better than most of the individual models (except LinearSVM) as it averages out the error from different models which results in slightly better accuracy.

Task	Macro F1-Score
CNN	0.5978
LSTM	0.6183
LSTM-A	0.6231
LSTM-CNN	0.5978

Table 5: Deep Learning Models

Among deep learning models, CNN model tries to encode sentence using 2D convolutions and this might result in some loss of information. Due to this reason, they have the lowest F1 score among the different models. LSTM model takes into account the sequential structure of the words as it processes word by word and it is able to perform much better than CNN.

LSTM-A model also takes into sequential structure of the words and it also helps the model focus on particular words that determine the emotion of a sentence through attention weights. This helps the model understand the significance of the words that pave way for a particular emotion and it enables this model to achieve the best performance among the different models.

LSTM-CNN model doesn't perform as expected. One of the possible reasons might be due to the low number of epochs. Another possible reason for this can be the loss of information from the encoded values due to the presence of 1D convolution kernels in the CNN part. This model was not run for more iterations as it was computationally expensive due to the complexity and number of layers.

The scores clearly indicate that the deep learning models perform much better than most of the traditional machine learning models. This shows that the sequential structure and position of the words in a sentence plays a key role in the overall emotion of the sentence and since deep learning models are able to capture this better than the count based traditional machine learning models, they perform better.

6 Conclusion

In this paper, we worked on Implicit Emotion Detection task (part of SemEval 2017) using different models, both traditional machine learning models and deep learning models. Due to the large labelled data that was available for training, the deep learning models performed better than the traditional machine learning models since they are able to better capture the sequential structure of the sentence. LSTM model with self attention outperforms the other models clearly indicating the effectiveness of using attention in a model and its impact.

7 Future Work

Due to computational constraints, the models were only trained for 5 epochs. We believe that the model can achieve better results if trained for more epochs. Secondly, the glove embeddings were used for pre-training the embedding layer weights. Word2Vec vectors, Doc2Vec vectors or FastText vectors can be used instead of this. Thirdly, the hyperparameters of all the models such as learning rate,number of layers,number of nodes,kernel dimension,kernel size etc. can be tuned to optimize the models and obtain better accuracy. Further, the models can use embeddings obtained from pre-training on tasks such as language modelling(ELMO-Net) and chat-bot con-

versations and this might capture the language space in a much better way and can eventually lead to better classification. We can also fine tune these embeddings to check the results in any improvement over just using frozen embeddings. Moreover, complex models such as transformer, BERT that performed much better on various NLP tasks can be used to model this problem.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 718–728.

Hassan Alhuzali, Mohamed Elaraby, and Muhammad Abdul-Mageed. ???? Ubc-nlp at iest 2018: Learning implicit emotion with an ensemble of language models .

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 579–586.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. pages 69–78.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM* 56(4):82–89.

Amel Fraisse and Patrick Paroubek. 2014. Twitter as a comparable corpus to build multilingual affective lexicons. In *The 7th Workshop on Building and Using Comparable Corpora*. pages 17–21.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Roman Klinger, Orphée De Clercq, Saif M Mohammad, and Alexandra Balahur. 2018. Iest: Wassa-2018 implicit emotions shared task. *arXiv preprint arXiv:1809.01083* .

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

Yang Liu and Xie Zhu. 2018. Short text sentiment classification based on feature extension and ensemble classifier. In *AIP Conference Proceedings*. AIP Publishing, volume 1967, page 020051.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Proceedings of AAAI*.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence* 29(3):436–465.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Jacopo Staiano and Marco Guerini. 2014. Depechemood: a lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605* .

Maite Taboada. 2016. Sentiment analysis: an overview from linguistics .

Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016a. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 225–230.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016b. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 2428–2437.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016c. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. pages 606–615.

Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. 2007. Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, pages 133–136.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*. pages 649–657.

8 Appendix

8.1 Contributions of Una

Una was responsible for developing, training, and validating/testing a simple LSTM model that would be used as one of the baselines for the more complex models. She chose to work on this particular model because LSTMs have been used for

speech recognition and her main research interest is speech perception. Since her background in Python and RNNs is not as strong as her team mates, she preferred to try to work on the model on her own first to make sure that she could do it. She worked on an initial version but that one had a different architecture than the others so later she switched it to match the others. Much of the code for her second version came from Arun because by that time he had already written it for his complex models. However, Una ensured that she understood the code and the linked packages. As far as the writing is concerned, Una collaborated on the literature review and also writing of the first half of the paper. She also reviewed the near-final draft of the paper.

8.2 Contributions of Ife

Ife was responsible for the baseline implementation of LSTM-A. She also worked on writing the latex file and editing the content

8.3 Contributions of Arun

Arun developed the traditional machine learning models. He worked on CNN model and LSTM+CNN model. Before building the LSTM+CNN model, he built the basic LSTM and LSTM with attention model and since these results were better than traditional models, they are reported. He also worked on writing the latex file and editing the content.