CS 4202

Research and Development Project

Emotion Analysis for Conversational Texts

Project Progress Report – Group 21

Supervisor - Dr.T.Uthayasanker,
Senior Lecturer,
Department of Computer Science and Engineering,
University of Moratuwa,
Sri Lanka.
(rtuthaya@cse.mrt.ac.lk)

Mentor - Mr.Yasas Senarath
Graduate Research Assistant at DataSEARCH,
University of Moratuwa,
Sri Lanka.
(wayasas.13@cse.mrt.ac.lk)

Team Members - V.Jarsigan (jarsigan.15@cse.mrt.ac.lk) – 150242N
(Group - 21) N.Piruntha (piruntha.15@cse.mrt.ac.lk) – 150418K
V.Vahesan (vahesan.15@cse.mrt.ac.lk) – 150648N

Table of Contents

Table of Figures

1. Introduction

Emotion analysis is the study of the variance of psychological and physiological states that are generated in human due to several external as well as internal changes. Conversational texts are the textual representation of the set of utterances that are used by two or more people to communicate among them. The basic idea behind this research project is to detect the emotions expressed through conversational texts.

The emotion analysis has a long history with several research been conducted on the topic with different perspectives such as text analysis, voice analysis, video analysis, multimodal analysis, etc. Among these, most of the research on emotion analysis of texts are focused on single utterances or tweets while a few of those are done on conversational texts with multiple utterances. We are working on to develop a deep learning-based solution to detect the emotions of conversational texts. This is to analyse each utterance of the whole conversations provided and classify them into three main types of emotions such as Happy, Sad and Angry.

It is evident that human nature is very complex and changes within a very short period. Thus, analysing such emotions is a hard task and its complexity increases further when considering the conversations because the emotion of a particular conversation may change after each utterance. Moreover, there could be several emotional ambiguities in the texts and the datasets maybe informal. We are working on to overcome all these challenges and provide an effective and efficient model to analyse the emotions in conversational texts.

2. Problem Statement

Statement

We are developing a more effective and efficient system that would recognize the emotions of conversational texts. Effectiveness of the system would be addressing the high accuracy of the results while the efficiency would ideally provide the best results using comparatively small data set.

Motivation

Analysing conversational tweets would be helpful in several important fields and sectors such as business, medical, cognitive computing and public services [1]. In the business sector, companies can use conversation analysis to understand the thoughts of the public about their products and to analyse their competitors' products [2],[3]. It will help them to improve their products and get a good market for them. Further, in the medical field, considering the psychology sector, emotion analysis would be helpful for the psychiatrists to mentor patients. Changes in the emotions of a patient according to various questions identified through this would help the psychiatrists to act accordingly. Moreover, these types of learning could be helpful to improve the cognitive computing applications like chatbots and smart homes as well [4].

3. Research Objectives and Project Outcomes

Research Objectives

The main objective of this research is to find a novel approach for Emotion Analysis of Conversational Texts. To achieve the main objective, we have to fulfil the following sub-objectives.

- Implementing a word embedding model that could provide more accurate results for comparatively small data sets.
- Developing an effective deep learning model for finding the emotions of conversational texts.
- Quantifying its impact using the benchmark provided in “EmoContext”, a competition by Microsoft.

Project Outcomes

The expected deliverables at the end of the project are as follows.

- A pre-processing model for conversational texts.
- A word embedding model to provide suitable results for low resource data.
- A deep learning model to find the emotions of the conversational texts.
- Research publications - based on the findings and improvements from the project.

4. Literature Review

Emotion analysis is one of the developing fields which was initiated as the emotion analysis for single tweet/ utterance. The paper titled as “DataSEARCH at IEST 2018: Multiple Word Embedding based Models for Implicit Emotion Classification of Tweets with Deep Learning” [5] is a good example for single tweet emotion analysis. The model mentioned in this paper consists of three main components. They are the pre-processor, feature extractor, and classifier. The combination of Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) [6] has been used as feature extractor while the Feed Forward Neural Network (FNN) is used as the classifier. They have worked with various types of word embedding models such as Twitter Word2Vec[7], Google News Word2Vec, wiki fastText, Glove, and wiki sub word fastText, and their combination to train the feature extractor to select the combination with the optimum performance among them. There are some limitations found in this system such as LSTM having a complex network and taking too much time, which is not good for evaluating small dataset.

Emotion analysis could be done not only for texts but also for videos as well. Interactive Conversational Memory Network (ICON) [2] is a model which has been designed to get the emotions in the conversational videos, especially the dyadic videos. This model consists of Multimodal Feature Extraction, Self-Influence Module, Dynamic Global-Influence Module, and Multi-hop Memory. The vector representation is generated by extracting the features from the text, audio, and video modalities. Here, simple CNN is used to extract the feature from the text, OpenSMILE [8] is used to extract the feature from audio and 3D-CNN is used to extract the features from the video. Self-Influence module consists of local Gated Recurrent Units (GRUs) to detect self-dependencies and the inter-personal module consists of global GRU to detect interpersonal dependencies. Multi-hop memory provides the predicted emotion by doing many memory read/write cycles. This model considers inter-personal dependencies while other models only focus on self-dependencies.

EMOTEX (Detecting Emotions in Text Messages) [4] is a model that could be used to analyse emotions of twitter messages. It consists of three main components namely, pre-processor, feature extractor, and classifier. 28 basic emotion keywords are used in this approach to get the labelled data. By using those keywords along with seed keywords of them and the wordnet sync set of them, labelled data is obtained.

Feature selection of EMOTEX is happening using four techniques such as Unigram Features [9], Emoticon Features, Punctuation Features, and Negation Features. Four different classifiers are used here including support vector machine, Naive Bayes, Decision Trees, and K-Nearest Neighbours for the classification function. One of the positives in this model is that the slang words have been handled by converting them as normal words with the removal of extra letters.

Sentiment and Semantic-Based Emotion Detector [10] is one of the recent models to detect the emotions in conversational texts. This model focuses emotions in two main parts such as sentiment analysis and semantic analysis. Sentiment analysis is done by using Sentiment Specific Word Embedding (SSWE) [11] and semantic analysis is done by Glove [12] and Fasttext [13]. This model has used a similar class of emotions and the dataset as ours.

Bert is a language representation model which can be easily fine-tuned with just one additional output layer to create state-of-the-art models for a huge range of tasks including sentiment analysis. It can bring a considerable improvement in classification performance under extremely small data in the task of text multi-classification. Bert could be used for our emotion analysis task by using the fine tuned model of that [15].

The fastText is a context-free word embedding model which is a library for efficient learning of word representations and sentence classification. There are many types of pre-trained models available in fastText, which can provide good outputs for the emotion analysis task [5].

The Context2Vec is introduced as an unsupervised word embedding model using bidirectional LSTM for efficient learning of generic context embedding of wide sentential contexts. When creating the embedding, it captures the sequence features in the context of the target word. This model works well while embedding target word and embedding context. This model is built based on the Word2Vec word embedding model. Both of Word2Vec and Context2Vec models learn context and target word representations at the same time, by embedding them into the same low-dimensional space. A much more powerful parametric model is used in Context2Vec to work more effectively in the sentential context [19].

DeepMoji is a word embedding model developed to learn the useful textual expressions used in sentiment analysis related tasks [20]. Using DeepMoji before sending the texts directly to the models related to social media sentiment analysis was proved to be much more effective because it deals more with emojis. DeepMoji uses two bidirectional LSTM layers together with an attention layer and a softmax layer to train the samples which have a higher number of emoji in them. This would help to bring out the emotions behind those emojis in particular. Thus, using this in the emotion analysis of conversational texts would yield better performance.

Universal Sentence Encoding [21] is an encoding representation to change the textual data into vector format, that specifically targets transfer learning to other NLP (Natural Language Processing) tasks. There are two different models available in TensorFlow hub for this such as transformer-based model and DAN (Deep Averaging Network) based model. It generates vector array of length 512 for the given text. For this representation, tokenization is not needed before feeding it to the encoder. Transfer learning from transformer-based sentence encoder usually performs better than DAN model. However, the DAN model is faster and simpler and could be better for some tasks. Moreover, DAN model is more suitable for small text conversation rather than the transformer type model.

5. Methodology

For our research topic, “Emotion analysis for conversational texts”, we had initially proposed a system as a solution based on the research papers we had read and existing systems analysed by us. But with the progress of the research, the methodology of the proposed solution has been modified according to the results and analysis of the experiments done with various models. However, the abstract view of the methodology has not been changed during this progress.

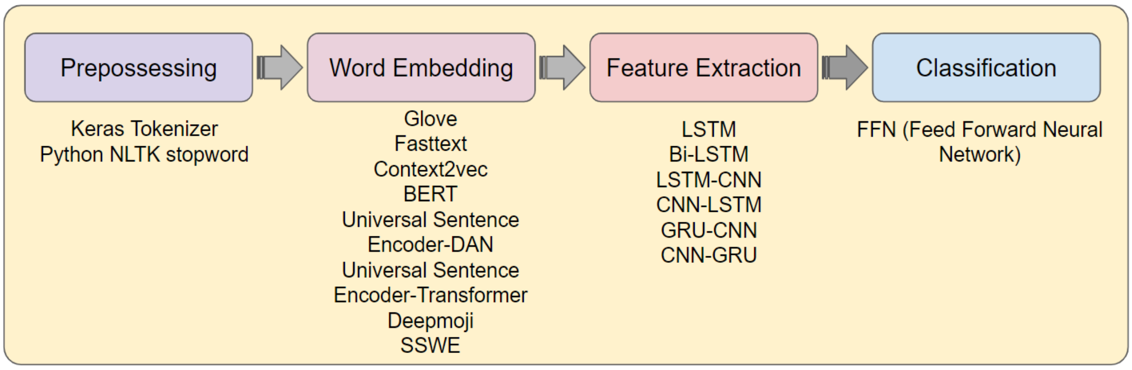


Figure 1- Experimental Models Tried

The high-level view of the system is explained first and that is followed by a further description of each component separately and deeply in this section. It provides a better idea about our methodology and way of our research.

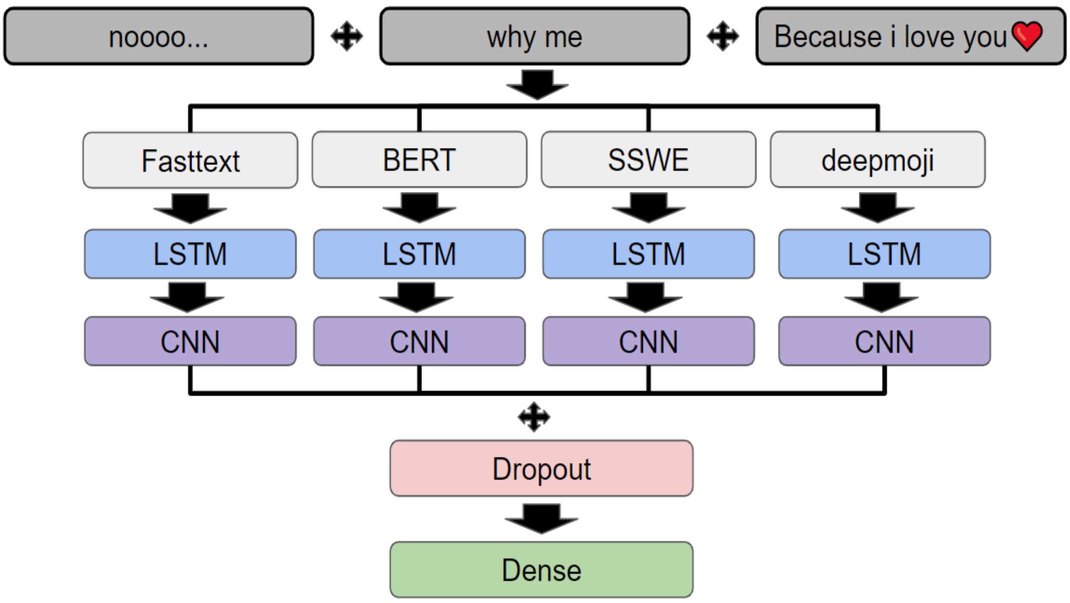


Figure 2 - Proposed Model Diagram

Our system mainly addresses the four main functionalities and those are completed by four separate components of the system. The main functionalities of the system are Pre-Processing, Word Embedding, Feature Extraction, and Classification. Each of these components contains sub functionality modules. In case of word embedding, we worked with contextual, context-free, sentiment and emoji type of word embedding to generate vector representations for our dataset. We have planned to combine these four different sets of embedding features and feed to the feature extraction process to observe all features from the dataset.

For feature extraction, we have tried LSTM (Long Short Term Memory Network), Bi-LSTM (Bidirectional Long Short Term Memory Network), LSTM-CNN (Long Short Term Memory Network-Convolutional Neural Network), CNN-LSTM, CNN-GRU (Convolutional Neural Network-Gated Recurrent Unit), GRU-CNN layers separately. We are working on these feature extraction layers with different embedding layers and other parameters accordingly. Therefore, we have not finalized the model layer yet as we are planning to use the combination of the models which provide optimum results. Finally, for the classification, we have tried using FNN (Feed Forward Neural Network) [5].

Next, each component would be considered and explained separately. An extra focus should be provided for dataset organization and pre-processing for the emotion analysis of the conversation type of texts. Each sample in the dataset contains three continuous utterances from the conversation between two people. For pre-processing we tokenized all the conversations using Keras Tokenizer. Moreover, all the words irrelevant to the emotion context are removed using Python NLTK stop word library. The conversation dataset has been fed into the model using various manners such as three utterances as one input, every three utterances as separate inputs and only last utterance as an input. After evaluating all those different methods, it was found that the three utterances as single input yields the optimum results for the classification than the other two methods.

The textual format data should be converted to vector format which could be understandable by the machine in order to be used with the deep learning technique. The technique used to convert the text to vector representation is known as word embedding. Here, we have tried with several word-embedding models as stated earlier, in order to optimize the results.

For context-free embedding, a Glove model pre-trained with 300 dimensions and a FastText model tuned by tweet data are used. These embedding models would help to find language semantic features. Further, for semantic and sentiment analysis, Universal Sentence Encoding (DAN), Universal Sentence Encoding Large (Transformer), BERT (Bidirectional Encoder Representations from Transformers) and Context2Vec models have been used. Moreover, we have planned to use SSWE (Sentiment Specific Word Embedding) for sentiment based analysis.

Nowadays conversations are more based on Unicode stickers (or Emoji). All of the global digital messaging systems such as WhatsApp, Viber, Messenger, and others are observed to be using Unicode stickers often. Thus, an extra focus should be provided to the stickers in conversations. Emoji2vec and DeepMoji pre-trained models could be handy in this as they give prominence to the stickers. Moreover, we have planned to give more weight on emoji features. We are still working on these different word embedding models with different feature extraction layers mentioned above to choose the optimum word-embedding model.

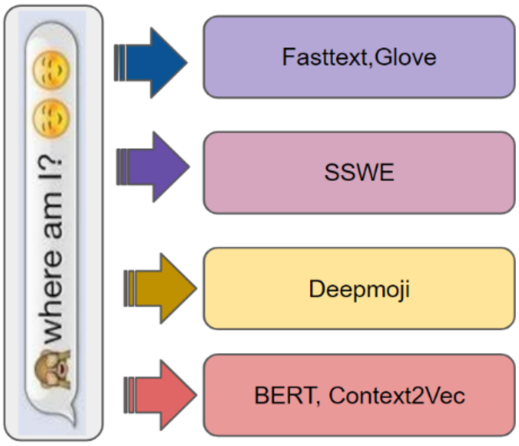


Figure 3 - Combined Word Embedding Model

After the processing in the word embedding model, vector representation of all the conversations would be obtained separately. Each of those vector data sets is provided as an input to the feature extractors. Here we have worked on different feature extractors that we mentioned earlier. We did not get good results by using combined models such as LSTM-CNN, GRU-CNN because of the model complexity results in overfitting on our small data set. Best result up to date has been obtained when using the Bi-LSTM model.
After the feature extraction, a set of feature-extracted vectors based on each word embedding would be obtained. Those can be categorized into three types namely sentiment based, semantic based and emoji based. All those three types of features would be combined together by concatenation operation. Here we provide high priority to emoji-based analysis and combine all those analysed feature vectors using concatenation operation. This would yield an effective final feature extracted vector which would be sent as input to the classifiers.

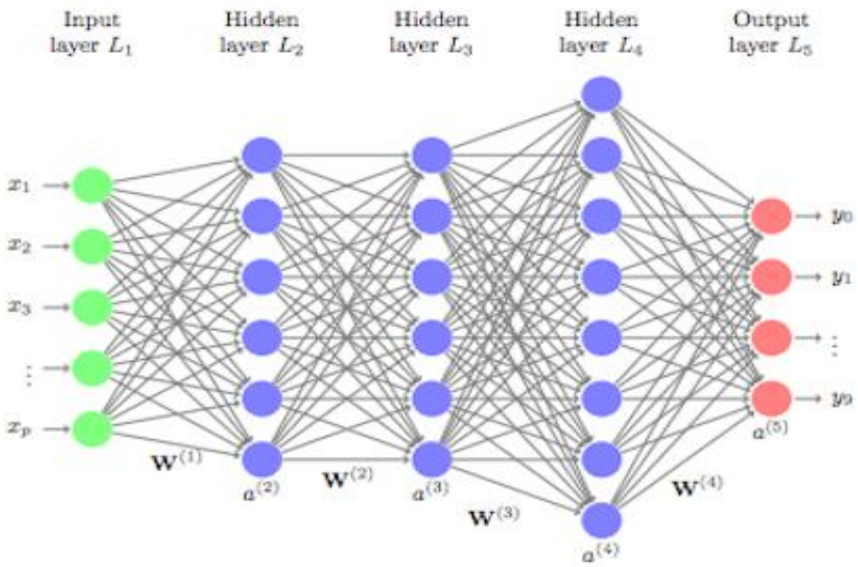


Figure 4 - Feed Forward Neural Network (FNN)

Finally, the effective feature extracted vector would be injected into the classifier where FNN is used for classification. It contains several layers of neurons which are categorized into three types as input layer, hidden layer, and output layer. Number of layers would be decided according to the context and necessity. Neuron is the basic unit of FNN and it contains specific evaluation function. Sometimes, this function is needed to be changed in order to obtain highly accurate results. When considering the high-level view of FNN, it gets a feature-extracted vector as the input and gives the percentage of our consideration as the output. In our case, we consider only the Sad, Happy, Angry and Others cases.

6. Results and Analysis

As we have used the dataset provided in the “EmoContext” competition conducted by Microsoft-Codalab, we are using micro F1 score, used in that competition as our evaluation criteria.

$$F1\mu = H(P\mu, R\mu) = \frac{2.P\mu.R\mu}{P\mu + R\mu}$$ (H:Harmonic mean)

$$Precision = P\mu = \frac{\sum TPi}{\sum(TPi + FPi)} \forall i \{Happy, Sad, Angry\}$$

$$Recall = R\mu = \frac{\sum TPi}{\sum(TPi + FNi)} \forall i \{Happy, Sad, Angry\}$$

- Initially, we had started working with the baseline code provided by the “EmoContext” competition conducted by Microsoft-Codalab. In that code, they have used Glove embedding and LSTM which yields 0.5791 micro F1 score.
- After analyzing the baseline code, we started improving that code. First, we changed the model by replacing the LSTM with the bidirectional LSTM. It gave the micro F1 value as 0.5599, which is lower than the initial one.
- So, we tried with different word-embedding models such as Bert, Contex2Vec and Universal sentence encoding-dan. The results obtained by using these embeddings are given below.

Word Embedding	**micro F1**
Bert	0.5313
Contex2Vec	0.6136
Universal sentence encoding-dan	0.5561

- Next, we tried to use different models such as CNN-LSTM, LSTM-CNN, GRU-CNN, and CNN-GRU with the above-mentioned word embeddings. The results obtained accordingly are mentioned in the table below.

Word Embedding	Model	micro F1
Bert	CNN-LSTM	0.6326
	LSTM-CNN	0.6044
	GRU-CNN	0.6115
	CNN-GRU	0.6163
	LSTM	0.5313
Context2Vec	CNN-LSTM	0.6018
	LSTM-CNN	0.6136
	LSTM	0.5561
	LSTM-CNN	0.5503
	LSTM	0.5437
Glove	CNN-LSTM	0.5944
	GRU-CNN	0.5587
	CNN-GRU	0.5506
	LSTM	0.5776

- After analysing the above results, there was no significant improvement found among them. Thus, we planned to use some embedding models that would give prominence to Emoji. For that, we selected DeepMoji embedding and tried another embedding called fastText which has 4 types. Results obtained for those are given in the table below.

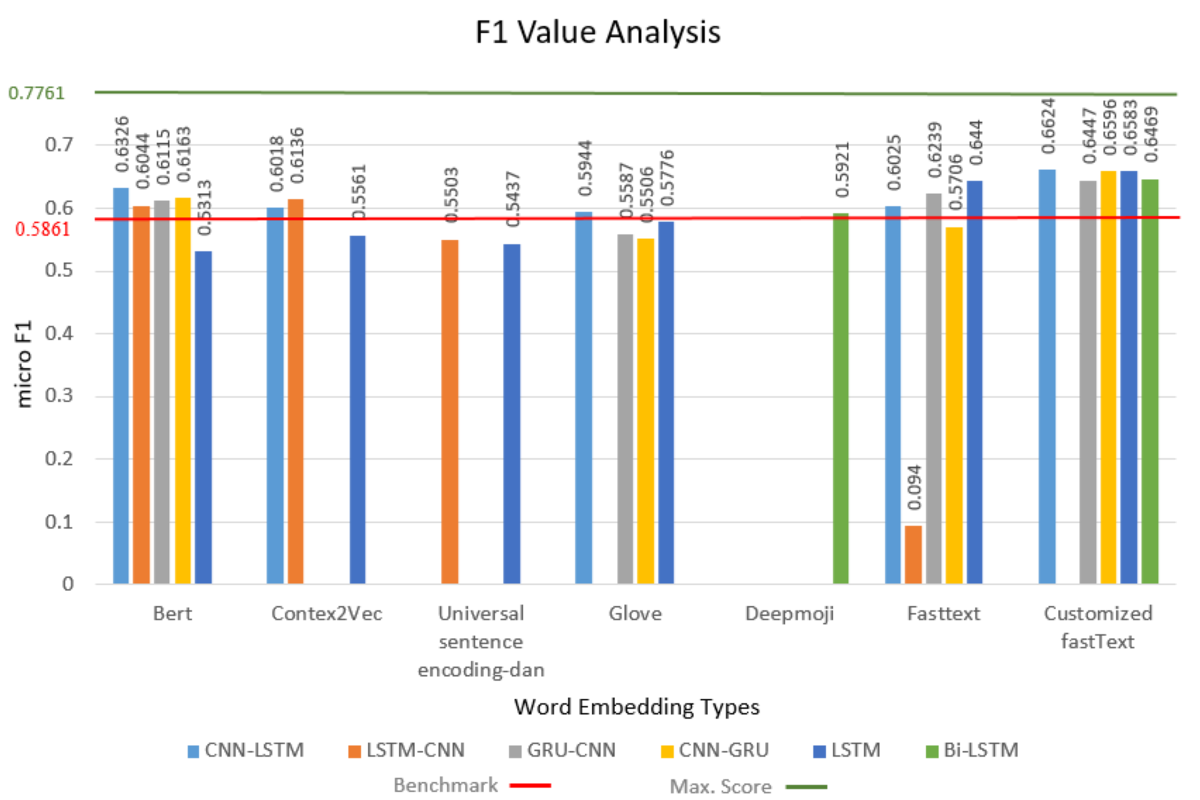
Word Embedding	Model	micro F1
FastText	CNN-LSTM	0.6025
	LSTM-CNN	0.0940
	GRU-CNN	0.6239
	CNN-GRU	0.5706
	LSTM	0.6440
DeepMoji	CNN-LSTM	In progress
	LSTM-CNN	In progress
	GRU-CNN	In progress
	CNN-GRU	In progress
	Bi-LSTM	0.5921

Fasttext Trained Type	Description	Micro F1
wiki-news-300d-1M	1 million word vectors trained on Wikipedia 2017	0.6149
wiki-news-300d-1M-subword	1 million word vectors trained with subword information on Wikipedia 2017	0.5954
crawl-300d-2M	2 million word vectors trained on Common Crawl (600B tokens).	**0.6440**
crawl-300d-2M-subword	2 million word vectors trained with subword information on Common Crawl (600B tokens).	0.6338

- After analysis, there was no noticeable improvement found. Thus, we tried to do an error analysis to find the problems. The details about the error analysis part are given in section 7 below.
- After that, we found a customized model of fastText embedding trained with tweet samples. The results obtained by using this embedding are given below.

Model	micro F1
LSTM	0.6583
Bidirectional LSTM	0.6469
CNN-LSTM	0.6624
LSTM-CNN	Inprogress
GRU-CNN	0.6447
CNN-GRU	0.6596

- The above fastText model gave a significant improvement in the micro F1 value. Thus, we are focusing on combining two models to improve the performance of the system now.
- The benchmark value provided in the "EmoContext" competition is 0.5861. We have beaten that already with our model and moving further with a comparatively better microF1 score of 0.6624.



7. Error Analysis

Using various word embedding models and different combinations of feature extraction models, we could not reach the expected level of micro F1 score. Thus, we tried doing error analysis for a sample set of data from our test dataset. For that purpose, we extracted a sample of 200 entries from the test dataset which should yield the following output.

Happy - 20

Sad - 19

Angry - 20

Other - 141

For the error analysis, we considered the DeepMoji, FastText, Context2Vec, and FastText (customized for tweets) models. The outputs we obtained in that manner are as follows.

		Actual Labels				
		happy	sad	angry	others	Total
Predicted Labels	happy	2	1	3	16	**22**
	sad	1	1	1	3	**6**
	angry	4	1	0	11	**16**
	others	13	16	16	111	**156**
	Total	**20**	**19**	**20**	**141**	**200**

Error analysis for deepmoji

		Actual Labels				
		Happy	Sad	Angry	Other	Total
Predicted Labels	Happy	15	0	0	5	20
	Sad	0	12	0	5	17
	Angry	0	1	17	6	24
	Other	5	6	3	125	139
	Total	20	19	20	141	200

Error analysis for fastText

		Actual Labels				
Predicted Labels		happy	sad	angry	others	Total
	happy	14	0	0	2	16
	sad	2	12	0	3	17
	angry	0	2	17	7	26
	others	4	5	3	129	141
	Total	20	19	20	141	200

Error analysis for contex2vec

		Actual Labels				
Predicted Labels		Happy	Sad	Angry	Other	Total
	happy	16	1	0	3	20
	sad	0	13	0	6	19
	angry	0	0	20	7	27
	others	4	5	0	125	134
	Total	20	19	20	141	200

Error analysis for customized fastText

	Deepmoji	Contex2Vec	fastText	Customized fastText
Happy	**0.9**	**0.3**	**0.25**	**0.2**
Sad	**0.95**	**0.35**	**0.35**	**0.3**
Angry	**1**	**0.15**	**0.15**	**0**
Other	**0.21**	**0.08**	**0.11**	**0.114**

Error rate

Using these outputs, we were able to find out that all the word embedding models we used to face the highest difficulties in finding the “Sad” emotion class. Further, the ‘Happy’ class and ‘Angry’ class follows them with a marginally small percentage of difference.

The conclusion from error analysis: -

- For the happy, sad and angry emotions customized fastText’s error rate is lower compared to other embeddings.
- For the other category, Contex2Vec has the lowest error rate compared to others.

8. Task breakdown

Month	Task breakdown	Status
January	• Literature review on similar models	• Completed
February	• Started working on proposal	• Completed
March	• Finalized the proposal • Started working with the baseline model	• Completed • Completed
April	• Tried a model with bidirectional LSTM • Tried with various word embedding models (BERT, Glove, Universal Sentence Encoding, Context2vec)	• Completed • Completed
May	• Tried with different models	• Completed
June	• Tried with DeepMoji • Tried with fastText • Error analysis	• In progress • Completed • Completed
July	• Worked with customized fastText • Combining word embedding models	• In progress • In progress

9. Future Tasks

- We have already tried the BERT model as an embedding model but, it can be used as a full model using transfer learning technique. We have planned to do this methodology instead of suggested methodology as an experiment.

- According to the error analysis result, we got low prediction emotion in Sad class. Therefore, we are planning to work on pre-processing level to improve the prediction by removing emotional ambiguous words.

- Combine DeepMoji with other semantic and sentiment representation.

- Use SSWE (Sentiment Specific Word Embedding) word embedding.

10. Conclusion

There are several research on emotion analysis of texts and a few regarding emotion analysis of the conversational text. Emotion analysis is one of the developing fields. In this project, we are working on to propose a more effective and efficient system that would recognize the emotions of conversational texts. We are mainly focusing on three types of emotions such as happy, sad and angry. We are using the dataset from “EmoContext”, a “CodaLab” competition conducted by Microsoft [16]. The benchmark value of the project will be the same as that given in the competition. We will be benchmarking the effectiveness of our developed model with existing models [2], [3], [4], [5],[10].

11. References

[1] F. Pascual, “Why and How Companies Should Use Sentiment Analysis - Level Blog” [online] Level Blog, Available at: https://www.northeastern.edu/levelblog/2018/08/02/companies-use-sentiment-analysis/ [Accessed 9 Jan. 2019].

[2] D. Hazarika, S. Poria, R. Mihalcea, E. Cambria and R. Zimmermann, "ICON: Interactive Conversational Memory Network for Multimodal Emotion Detection", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2594--2604.

[3] D. Hazarika, S. Poria, A. Zadeh, E. Cambria, L.P. Morency, and R. Zimmermann, "Conversational Memory Network for Emotion Recognition in Dyadic Dialogue Videos", in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 2122–2132.

[4] M. Hasan, E. A. Rundensteiner, and E. Agu, “EMOTEX: Detecting Emotions in Twitter Messages”, in *ASE BIGDATA/SOCIALCOM/CYBERSECURITY Conference*, Stanford University, May 27-31, 2014

[5] Y. Senarath and U. Thayasivam, "DataSEARCH at IEST 2018: Multiple Word Embedding based Models for Implicit Emotion Classification of Tweets with Deep Learning", in *Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, Brussels, Belgium, 2018.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks”, in *ICCV*, 2015, pp. 4489–4497.

[7] O. Melamud, J. Goldberger, and I. Dagan, “context2vec: Learning Generic Context Embedding with Bidirectional LSTM,” in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 51–61.

[8] F. Eyben, M. Wöllmer, and B. Schuller, “Opensmile,” in *Proceedings of the international conference on Multimedia - MM ’10*, 2010, p. 1459.

[9] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision”, 2009, pp. 1–12.

[10] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, “Understanding Emotions in Text Using Deep Learning and Big Data,” *Comput. Human Behav.*, vol. 93, pp. 309–317, Apr. 2019.

[11] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu and B. Qin, "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification", in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Baltimore, Maryland, 2014, pp. 1555–1565.

[12] J. Pennington, R. Socher and C. D. Manning, “Glove: Global vectors for word representation.”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[13] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, “Bag of tricks for efficient text classification”, *EACL*, 2017.

[14] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bošnjak, and S. Riedel, “emoji2vec: Learning Emoji Representations from their Description,” Sep. 2016.

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

[16]"CodaLab - Competition", *Competitions.codalab.org*, 2019. [Online]. Available: https://competitions.codalab.org/competitions/19790#learn_the_details. [Accessed: 17-Jan- 2019].

[17] A. Suarez, D. Albakour, D. Corney, M. Martinez and J. Esquivel, "A Data Collection for Evaluating the Retrieval of Related Tweets to News Articles", 2018.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, “Distributed representations of words and phrases and their compositionality*”, Advances in neural information processing systems*, 2013, pp. 3111–3119.

[19] O. Melamud, J. Goldberger and I. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM", *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning,* 2016.

[20] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm", *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[21] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. John, N. Constant, M. Guajardo-Céspedes, S. Yuan, C. Tar, Y. Sung, B. Strope and R. Kurzweil, “Universal Sentence Encoder”, 2018.