AFFECT LEVEL OPINION MINING OF TWITTER STREAMS

Wijesuriya Arachchige Yasas Pramuditha Senarath

188005H

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science and Engineering

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

July 2019

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters thesis/Dissertation under my supervision.

Signature of the Supervisor: Date:

ACKNOWLEDGEMENTS

I would first like to express my gratitude to my supervisor Dr. Uthayasanker Thayasivam and other advisers from DataSEARCH center who always shared their wisdom and experience to make this research study success. They constantly steered me towards right direction whenever they though I needed it. This research was supported by DataSEARCH center at University of Moratuwa, Sri Lanka.

I am sincerely grateful for Dr. Surangika Ranathunga and Dr. C.R.J. Amalraj for reviewing this research study at progress reviews and providing their valuable suggestions to improve the project outcomes.

I would also like to thank Mr. Kumaran Kugathasan for helping me to coordinate research activities. Further, I thank both academic and non-academic staff of Department of Computer Science and Engineering for giving their fullest support towards this project.

Last but not least, I would like to express my gratitude to all the members of DataSEARCH center of Department of Computer Science and Engineering for their constant monitoring and invaluable suggestions to make this study a success.

ABSTRACT

Affect Level Opinion Mining of Twitter Streams

Twitter is a social media platform which is used by millions of users to express their opinions freely. However, it is almost impossible to analyze the opinion manually due to the sheer number of Tweets generated per day. Therefore, automated analysis of emotions in Tweets, which is also known as affect level opinion mining in the literature is crucial. Emotion analysis in this study is performed at two levels: Emotion Category Classification and Emotion Intensity Prediction.

One key challenge in identifying emotion categories is the presence of implicit emotions. This study introduces a model that enables reuse of the same deep neural network architecture with different word embeddings for the extraction of different features related to implicit emotion classification. We presented this model at 9^{th} Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA-2018). Our system was ranked among the top ten systems (8^{th}) amidst constrained corpus usage. Our implicit emotion classifier outperformed the baseline system by more than 8%, achieving a 68.1% macro F1-Score.

We solved the emotion intensity task with transfer learning techniques. Among the models used in transferring features were a sentiment classifier, emotion classifier, emoji classifier and emotion intensity predictor. Our transfer learning based intensity predictor outperformed existing best in two out of four emotions. We were able to achieve an average Pearson score of 79.81%. Additionally, we propose a technique to visualize the importance of each word in a tweet to get a better understanding of the model.

Finally, we developed a web-platform that utilizes our emotion analysis models to summarize and view the opinion of a group of tweets.

Keywords: Emotion Classification; Emotion Intensity Prediction; Sentiment Analysis; Opinion Mining;

LIST OF FIGURES

Figure 1.1	An emotion intensity example.	2
Figure 2.1	Plutchik's wheel of emotions	6
Figure 3.1	Machine Learning Hierarchy	13
Figure 3.2	An isolated neuron	13
Figure 3.3	A three layer Feed-forward Neural Network	14
Figure 3.4	Convolution operation on a matrix	15
Figure 3.5	Unrolled RNN Structure	15
Figure 3.6	Common RNN module types	16
Figure 3.7	Different Settings of Transfer Learning	17
Figure 3.8	Transfer learning with Fine tuning	18
Figure 3.9	Transfer learning with Fixed Model	19
Figure 3.10	Cross validation iterations	21
Figure 4.1	Approach used in this study for emotion analysis	22
Figure 4.2	Overview of the implicit emotion classification architecture	24
Figure 4.3	FNN used as as final implicit emotion classifier	25
Figure 4.4	High-level LSTM-Conv Network Architecture	26
Figure 4.5	Overview of emotion intensity prediction architecture	28
Figure 4.6	Recurrent-Convolutional Neural Network	29
Figure 4.7	Emotion Intensity Regression - EITL Module	30
Figure 6.1	High-level Architecture of Emotion Visualization/Summarization Architecture	41
Figure 6.2	Emotion visualization and summarization platform	42
Figure 6.3	Individual emotion intensity visualization with the visualization platform	42

LIST OF TABLES

Table 1.1	Example Tweets classified according to emotion	2
Table 2.1	Results for systems evaluated on SemEval-2018 Task 1	7
Table 2.2	Comparison of system results for emotion annotations in [1]	8
Table 2.3	Emotion Intensity models at WASSA-2017	9
Table 2.4	Emotion Intensity regression models at SemEval-2018 Task 1	11
Table 3.1	Transfer Learning Approaches	16
Table 4.1	Embedding Models used in Experiments	26
Table 5.1	Emotion words used when collecting Tweets	32
Table 5.2	Distribution of the Implicit Emotion Dataset	32
Table 5.3	Network Parameters for LSTM-CNN	32
Table 5.4	Network parameters for FNN	33
Table 5.5	Evaluation of LSTM-CNN for different word embeddings	34
Table 5.6	Results of FNN for best feature combinations	34
Table 5.7	The number of tweets in the SemEval-2018 Affect in Tweets Dataset	36
Table 5.8	Model and training hyper-parameters for ECCU and EIPU models	37
Table 5.9	Parameters used for training EITL model	37
Table 5.10	Performance scores for ECCU compared with the benchmark systems	38
Table 5.11	Performance scores of emotion intensity prediction model	38
Table 5.12	Examples for word level importance heat-map visualizations	39

LIST OF ABBREVIATIONS

ABBR	Abbreviation
API	Application Programming Interface
ML	Machine Learning
FNN	Feed-foreword Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory Network
SVM	Support Vector Machine
SVR	Support Vector Regression

TABLE OF CONTENTS

Declaration of the Candidate and Supervisor i

Acknowledgement ii

Abstract iii

List of Figures iv

List of Tables v

List of Abbreviations vi

Table of Contents vii

1 Introduction 1

1.1 Problem 3

1.2 Motivation 3

1.3 Research Objectives 3

1.4 Contributions 4

2 Literature Survey 5

2.1 Emotion Classification 5

2.2 Emotion Intensity Prediction 8

3 Background 12

3.1 Artificial Neural Networks 12

3.2 Neural Network Layer Types 13

3.2.1 Dense Layer 13

3.2.2 Convolutional Layer 14

3.2.3 Recurrent Layers 14

3.3 Transfer Learning 15

3.3.1 Transfer Learning with Fine-tuning 17

3.3.2 Transfer Learning with Fixed Model 18

3.4 Word Embedding 19

3.4.1 Word2vec 19

3.4.2 Glove 19

3.4.3 FastText 20

3.5 Cross Validation 20
4 Methodology 22
4.1 Implicit Emotion Classification 22
4.1.1 Tweet Pre-processor 23
4.1.2 Models 24
4.1.3 Feature Extraction 25
4.2 Emotion Intensity Prediction 27
4.2.1 Tweet Pre-processor 27
4.2.2 Models 27
5 Evaluation 31
5.1 Implicit Emotion Classification 31
5.1.1 Dataset 31
5.1.2 Experimental Setup 32
5.1.3 Results 33
5.1.4 Discussion 35
5.2 Emotion Intensity Prediction 35
5.2.1 Dataset 35
5.2.2 Experimental Setup 36
5.2.3 Results 36
5.2.4 Discussion 38
6 Twitter Emotion Analysis Platform 41
7 Conclusion 43
7.1 Future Work 44
References 45

Chapter 1

INTRODUCTION

Social Media has already penetrated almost every part of the world and has become a major source of information for many. The users of social media can range from the general public to large organizations and even governments. The general public uses it to share their life activities and opinions about different topics with their family and friends. Organizations may use it to promote their products and services.

Twitter micro-blogging platform is one of the most popular social media platforms on the internet. Though a single Tweet is constrained by the number of characters, opinion of the public on some topic can be determined using a large number of twitter posts (Tweets). Tweets are publicly available. Therefore, one can collect them directly from the Twitter platform using the tools provided by Twitter (Twitter API [1]). Furthermore, determining the opinion from the short text is a challenging task. Consequently, the Twitter platform is a worthy source of social media content for research. The term "opinion" is defined as "a view or judgment formed about something, not necessarily based on fact or knowledge" . Many studies consider emotions as a direct determinant of opinion [2]. Accordingly, emotions can be analyzed to be used in various applications such as decision making, product enhancement, etc. However, it is almost impossible to analyze emotions in a bulk of social media posts manually. Computational analysis of emotions in social media posts such as Tweets has, therefore, become a priority for the research community in recent years.

[1] https://developer.twitter.com/en.html

Table 1.1: Example Tweets classified according to emotion

Emotion	Tweet	
Joy	Sharing first acrylic painting of Lord Buddha on this auspicious day	(1)
	The #happiness and #joy from #harvesting these #beautiful #tomatoes and #eggplant from #mygarden	(2)
Sadness	All you did was drown my heart in pain	(3)
	The new episode of game of thrones.. disappointed	(4)
Anger	Tired of scams model agencies ☹	(5)
	Beware, I'm not in my greatest mood today.. angry	(6)

Table 1.1 illustrates some randomly selected tweets containing emotions. Tweet (1) shows the opinion of the author on a special day of a religion. Next, tweet (2) indicates the happiness of the author of the garden harvest. Tweets (1, 2) are typical examples for emotionally happy tweets. Words such as "pai" and "disappointed" in tweets (3) and (4) are indicators of sadness. Emoji in tweet (5) and hashtag ("#angry") in tweet (6) indicates the author's anger. Emoji, hashtags and individual words may not always reflect the true emotional state of the content, however they may contribute in determining the overall emotion of a tweet.

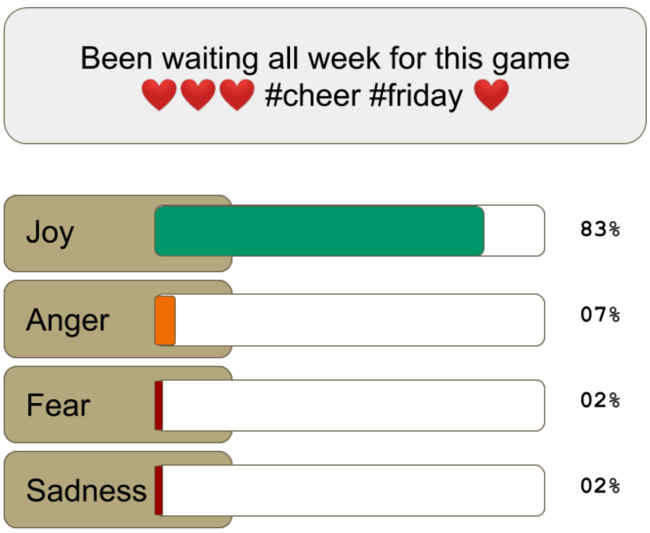


Figure 1.1: An emotion intensity example.

It is possible to determine emotions in tweets at different levels. The most primal procedure of emotion identification is polarity analysis. More specifically, identification of positive, negative, or neutral nature of the affect in a tweet. A more sophisticated technique to identify opinion is by identifying emotion categories associated with a tweet. A finer level of emotion identification is to associate each emotion with a value showing the emotion intensity. Figure 1.1 illustrates a sample Tweet with associated emotion intensities. Each bar in Figure 1.1 represents the degree/intensity of emotion expressed in that sample Tweet.

1.1 Problem

The existing state of the art techniques to identify emotion lack the sufficient accuracy and have limited capabilities to perform accurate prediction. List of limitations of existing approaches:

- Contextual Meaning: Identify emotions/ emotional intensity when complex relationships between words are observed.
- Implicit Emotion: Tweets may not always convey the feeling using explicit emotional words like sad, angry happy etc.
- Emotion Intensity: Emotion may be expressed at different level of intensity providing importance to some documents than other.

1.2 Motivation

Many natural language applications in social media analytics can benefit from knowing both the category and intensity of emotion in a Tweet. For example, a commercial customer satisfaction system would prefer to focus first on instances of significant frustration or anger, as opposed to instances of minor inconvenience. However, existing approaches lack the accuracy or have limited capabilities to accomplish this task.

1.3 Research Objectives

The main objective of this research is to identify and develop a mechanism to extract emotional information from Tweets and to summarize them for a set of

Tweets. Our sub-objectives in this research are as follows:

- to develop a system to predict emotion categories existing in a Tweet under different constraints
- to devise a methodology to identify intensity of a given emotion in a Tweet
- to identify a suitable method to summarize emotion information in Tweets and to visualize them

1.4 Contributions

Following contributions were made during this study;

- Introduce a novel architecture for emotion classification that enables reuse of the same neural network architecture for extraction of different features.
- Apply our novel architecture for implicit emotion emotion classification that ranked in top ten (8) in WASSA 2018 Shared Task for Implicit Emotion Classification [3].
- Introduction of simpler but effective models for emotion classification and intensity prediction.
- Apply state-of-the-art interpretation models to visualize and explain deep models for emotion intensity prediction.
- Develop an open-source platform for end-to-end emotion analysis for Tweets[2].

[2]https://github.com/ysenarath/opinion-framework

Chapter 2

LITERATURE SURVEY

Opinion mining is a renowned sub-field of Natural Language Processing (NLP) that deals with the identification of opinion orientation from user-generated textual content [4]. Research areas in opinion mining include polarity classification, sentiment intensity, emotion classification, and emotion intensity prediction [5, 6, 7, 8].

Polarity classification is extensively studied in the past decade [5, 9, 10, 11]. It deals with the identification of the positive, neutral, or negative nature of a document. Additionally, sentiment can be measured using discrete value indicating sentiment level or a real value indicating the sentiment intensity [6, 7]. However, polarity detection fails at identifying finer details about the opinion expressed in the text. We could use emotions as a finer and detailed alternative to polarity detection.

Several psychological studies have identified sets of discrete emotions [12, 13]. Plutchik's wheel of emotions indicated in Figure 2.1, provides a set of basic emotions, variations of emotion with intensity and compound emotions formed by mixing primary emotions. Eight basic emotions identified in the Plutchik's wheel of emotions are Anger, Anticipation, Disgust, Joy, Fear, Sadness, Surprise, and Admiration. However, Ekman [12] identifies only six basic emotions: Anger, Disgust, Fear, Joy, Sadness, and Surprise.

2.1 Emotion Classification

Emotion classification deals with assigning a set of emotions related to a document from a predefined set of discrete emotions such as Plutchik's basic emotions. Earliest work on emotion classification was based on sentiment lexicons and manually created features [8, 14]. A sentiment lexicon is a list of phrases or words with associated sentiment orientation [15]. Liu et al. have utilized a database contain-

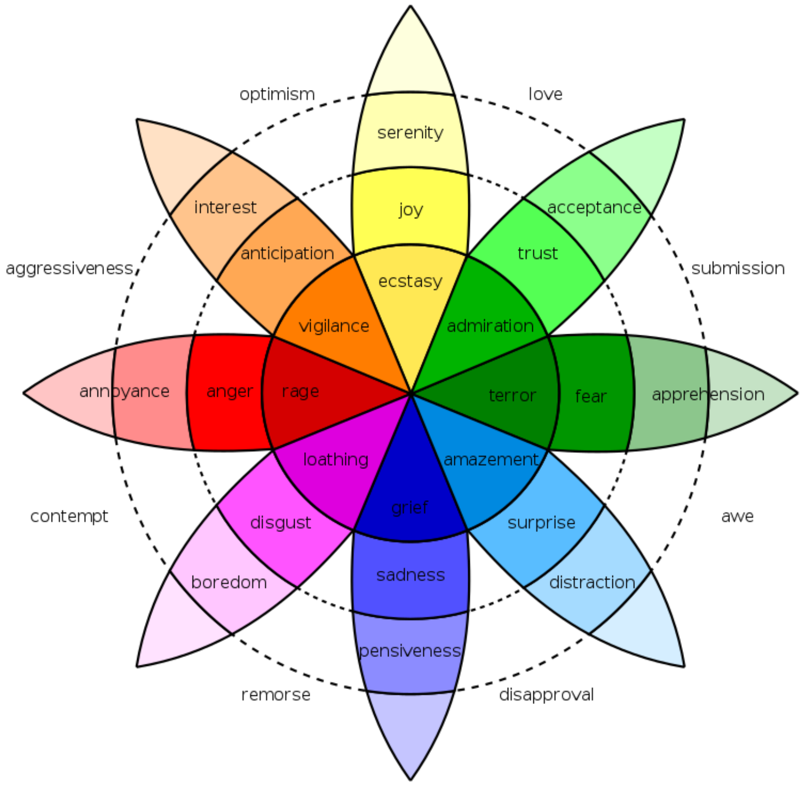


Figure 2.1: Plutchik's wheel of emotions
Source: https://de.wikipedia.org/wiki/Robert_Plutchik

ing background knowledge to identify the emotions in short text [8]. However, the scalability of this approach is less since it depends on predefined emotion lexicons and rules. Alm et al. [14] have introduced an emotion corpus (Alm's corpus) containing sentences in 185 children stories. Each annotator's task was to mark sentences with one of the extended set of basic emotions. However, in Figure 2.1, we can identify that emotions are not strictly discrete. It is possible to observe multiple emotions at once. Therefore, Alm's corpus can be inaccurate when there is more than one emotion in a sentence. Contrast to Liu et al. [8], Alm et al. [14] has used machine learning model to identify emotions.

Mohammad et al. [16] introduced their multi-task twitter emotion corpus as a competition at the International Workshop on Semantic Evaluation (SemEval) 2018. Emotion classification (E-c) was one task that was introduced. They have considered emotions as a multi-label classification problem. Therefore, multiple

Table 2.1: Results for systems evaluated on SemEval-2018 Task 1: Emotion Classification dataset.

Model	Features	Evaluation Results (%)		
		Acc.	Micro F1	Macro F1
NTUA-SLP [17]	Transfer weights from - Neural network trained on Semeval-2017 Task 4A dataset [18]	**58.8**	**70.1**	52.8
TCS Research [19]	Lexicon Features Transfer features from - Sentiment Neuron [11] - Multi-word-embedding Bi-LSTM Attention Neural Network Model	58.2	69.3	**53.0**
PlusEmo2Vec [20]	Transfer features from: - DeepMoji Model [21] - Bi-LSTM classifier trained to predict emoji clusters Emotional Word Vectors (EVEC) Tweet specific features Emotion correlation features - Classifier Chain - Regressor Model	57.6	69.2	49.7

emotions may be present in a single document. Table 2.1 shows the summary of the best systems using dataset provided in [16].

[17, 19, 20] were all mainly based on transfer features. This has enabled the use of models trained to solve one task to be used in solving another. Although [19] try to incorporate multi-word-embedding models to create single word representation, the approach however is using incompatible mixture of word embeddings. For example, emoji2vec [22] is expected to be used with Google News word2vec [23] since emoji2vec is built on top of Google News word2vec, however they use it with common crawl glove embedding with compatible size embeddings. Neither they provide a reason for using glove embedding instead of word2vec nor experimentally show that this method is better than using recommended word2vec.

Majority of the models on emotion classification relies on emotion word information [8, 14, 16]. This is possible since emotion words such as "happy", "angry" and "sad" are explicitly mentioned in the text. However, emotion words may not appear in the document at all. Therefore explicit emotion models may not be able to predict implicitly expressed emotions. [24] has produced a dataset on implicit emotion classification and released it as a competition task at Workshop on Com-

Table 2.2: Comparison of system results for emotion annotations in [1]

Model	Pearson Correlation (%)						Average
	Anger	**Disgust**	**Fear**	**Joy**	**Sadness**	**Surprise**	
SWAT [25]	24.51	**18.55**	32.52	**26.11**	38.98	11.82	25.42
UPAR7 [26]	**32.33**	12.85	**44.92**	22.49	**40.98**	**16.71**	**28.38**

putational Approaches to Subjectivity, Sentiment, and Social Media (WASSA) 2018.

[24] depended on sentence patterns to automatically label the dataset. For example,

"I feel sad because I'm alone at Home" — Sentence (1)

If we remove word sad from the Sentence (1) indicated above, we get only the reason for his/her sadness. Therefore, we can label this sentence with emotion "sad" even after removing the explicit mention of word "sad" in the text. The resulting sentence is "I feel [emotion] because I'm alone at Home". Likewise sentence patterns are used in [24] to automatically label the dataset.

2.2 Emotion Intensity Prediction

The first recorded work on automated emotion intensity detection is Strapparava et al. [1]. It was introduced as a SemEval task to identify emotions and positive/negative nature (valance) in news headlines. Annotators were provided a web interface to indicate the degree of emotion for each news headline using slide bars. 250 headlines were annotated to be used as development dataset with another 1000 headlines were used to test the results. The results of top two teams are compared in Table 2.2. An obvious observation is that these systems lack the performance in predicting the correct fine-grained emotion in comparison to human annotations. This can be attributed to the low inter-annotator agreement Pearson correlation scores described in [1].

Table 2.3: Emotion intensity models at WASSA-2017. Tick (✓) mark is used to indicate the use of given model or feature in the study.

Model	**Feature Types**			**Regression Model**					**Average Pearson Score (%)**
	WE	**AL**	**NG**	**N**	**R**	**S**	**L**	**E**	
Prayas [30]	✓	✓		✓				✓	74.7
IMS [31]	✓	✓		✓	✓				72.2
SeerNet [32]	✓	✓			✓		✓	✓	70.8
UWaterloo [33]	✓	✓	✓	✓		✓	✓	✓	68.5
IITP [34]	✓	✓					✓		68.2

Neviarouskaya et al. [27] uses rule based approach to derive the emotion category and the emotion intensity of a corpus containing sentences from blog entries . However, they have not evaluated the estimated emotion intensity values.

A typical challenge in the annotation process used by [1] is the inconsistency. It is highly unlikely that two annotators annotate the same emotion intensity value for the same sentence. [28] provided a solution to this by asking the annotators to choose the emotionally most intensive (best) and least intensive (worst) tweets from four tweets selected from the corpus multiple times. This method of annotation is called 'Best - Worst Scaling (BWS)' [29]. It has enabled them to create a reliable dataset with higher consistency since it is likely for two people to agree on similar best and worst instances from given set of options than providing a independent value without looking at other instances. [28] has introduced their dataset in a Shared Task on Emotion Intensity at WASSA-2017. Twenty one teams participated Shared Task on Emotion Intensity WASSA-2017. Table 2.3 summarizes the top five systems and their final evaluation results. Feature types indicated by WE, AL, and NG refers to word embedding (word represented as vector), affective lexicons and N-grams accordingly. Regression models identified by N, R, L, S, and E refers to Neural Networks, Random Forest (RF), linear regression, Support Vector Regression (SVR) and ensemble.

Top performing system [30] uses a combination of multiple neural network techniques in the final model. Fist is a Feed-forward Neural Network (FNN) which takes in word2vec [23] and affect lexicon features to predict the intensity for a given emotion.They trained an individual FNN for each emotion. A drawback of this approach is that this method cannot be used to share information between emotions. Second technique is a neural network trained using Multi Task Learning (MTL) where they try to improve the performance by combining datasets for each individual emotion. Next, they use sequence model using (Convolutional Neural Networks) CNNs and (Long-Short Term Memory) LSTM Networks to predict the intensity of emotions. [30] provides the final prediction by taking the weighted average of the individual techniques. Although [30] studies the effects of using different neural network models extensively, they do not try to identify the possibility of using features learned in other tasks such as emotion classification in to emotion intensity prediction.

In contrast to [30], [31] combines neural network approach with traditional machine learning techniques. [31] extends the vocabulary of affect lexicons by training a FNN to predict the affect intensity using word embedding features. Then they use affective lexicon features generated using above model for training the sentence level model. Moreover, they use predictions of CNN-LSTM neural networks as additional feature generators for the final prediction. Finally, they concatenate all the features and train a random forest regressor to obtain final emotion intensity.

In comparison to [30, 31], [33] does not use neural networks in their approach. [33] has experimented with SVR, RF, AdaBoost and Bagging Regressor. After validation they found that AdaBoost with XG-Boost as base estimator outperforms other approaches. Although, this method allows relatively easy interpretation, it fails to capture complex features such as the sequence of the words (order). Another drawback of this approach is that we have to manually engineer the correct features.

The latest dataset in emotion intensity prediction is released as SemEval-2018 Task 1: Affect in Tweets by Mohammad et al. [16]. [16] extends work of [28] by increasing the number of annotated tweets and and including other tasks such as emotion classification and valance intensity. Table 2.4 summarizes the methods and results of top performing systems.

A common observation from Table 2.4 is the use of transfer learning in all top

Table 2.4: Emotion Intensity regression models at SemEval-2018 Task 1: Affect in Tweets. Model types indicated by characters N, R, X, S, E are Neural Network, Random Forest, XG-Boost, SVR and Ensemble Methods respectively.

Model	**Features**	**Model Type**					**Average Pearson Score (%)**
		N	R	X	S	E	
SeerNet [32]	Transfer features from: - DeepMoji Model - Skip Thoughts Vector - Sentiment Neuron - EmoInt		✓	✓		✓	**79.9**
NTUA-SLP [17]	Transfer weights from - Neural network trained on Semeval-2017 Task 4A dataset	✓					77.6
PlusEmo2Vec [20]	Transfer features from: - DeepMoji Model - Bi-LSTM classifier trained to predict emoji clusters Emotional Word Vectors (EVEC) Tweet specific features	✓			✓	✓	76.6

three models. Compared to the models in [28], models in [16] leverages the use of domain adaptation/transfer learning. Although [32] uses features from pre-trained neural networks, their methodology was based on traditional machine learning techniques. They use XG-Boost and random forest regression to predict the emotion intensity from feature vector collected from a pretrained model. Finally, they combine the results of each individual model using ensemble methods. This approach fails to identify the relationships between features in pre-trained models since feature vectors obtained from each pretrained model is used separately to train individual models.

Chapter 3

BACKGROUND

This section will describe the important techniques and tools that are used in this study.

3.1 Artificial Neural Networks

Figure 3.1 represents the typical classification hierarchy of Machine Learning (ML) techniques. In this study, classification and regression techniques were used to classify emotions and predict emotion intensity respectively. Majority of models were obtained after training deep neural networks.

Artificial Neural Networks (ANNs) are set of algorithms modelled after biological neural networks. Figure 3.2 illustrates an neuron, the basic unit of a neural network. X1 to Xn indicates input to that given neuron while "f" indicates the activation function. W0 is the bias of the neuron.

Figure 3.3 represents a three-layer neural network. Neurons in this neural network are indicated by circles. Input neurons are prefixed with "I" in the diagram. We identify that as the input layer. Neurons prefixed with "O" forms the output layer. Layers in between input and output layers are referred to as hidden layers. Connections in between the neurons provide the flow of data from output of one neuron to the input of another.

Deep Neural Network is generally referred to artificial neural networks with more than two layers. Figure 3.3 represents a deep neural network since it consists a hidden layer.

When designing a neural network we have to choose set of hyper-parameters. This includes the number of layers we are using, type of layer, type of activation in neurons and number of neurons in each layer. This has to be carefully selected since more complex neural networks can easily overfit the model while simple networks may under-perform.

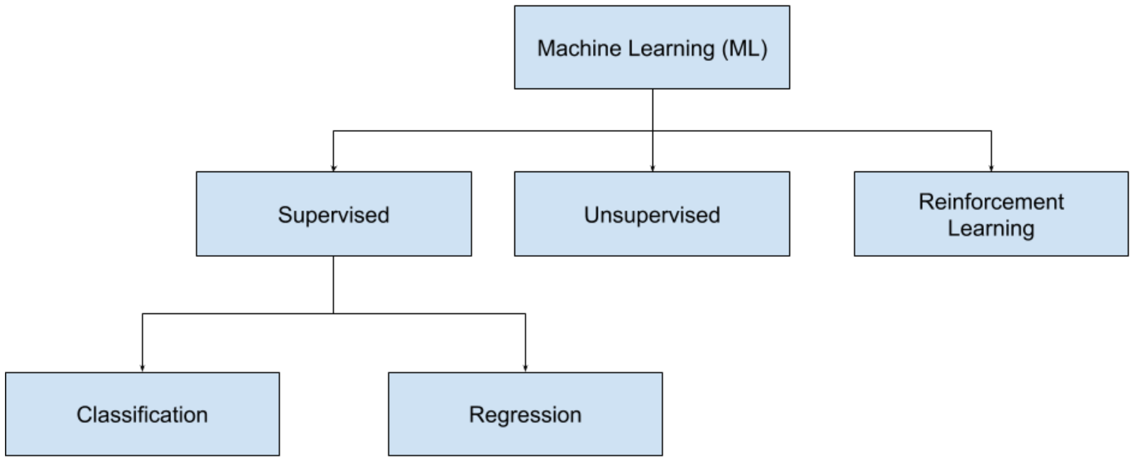


Figure 3.1: Machine Learning Hierarchy

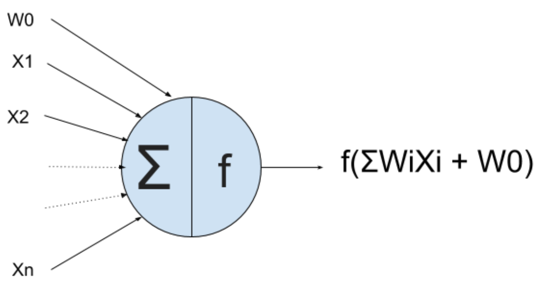


Figure 3.2: An isolated neuron

There are different types of ready-made neural network layers that we can use in our implementations. These include Dense Layer/ Fully Connected Layer, Convolutional Layer with Pooling, Recurrent Layers such as Long-short term memory networks (LSTMs) and Gated Recurrent Units (GRUs).

3.2 Neural Network Layer Types

This section describes different neural network layer types that we have used in our methodology.

3.2.1 Dense Layer

Dense layer has connections from every neuron in previous layer to every neuron in dense layer. Important tunable parameters of dense layer are the number of neurons and activation function.

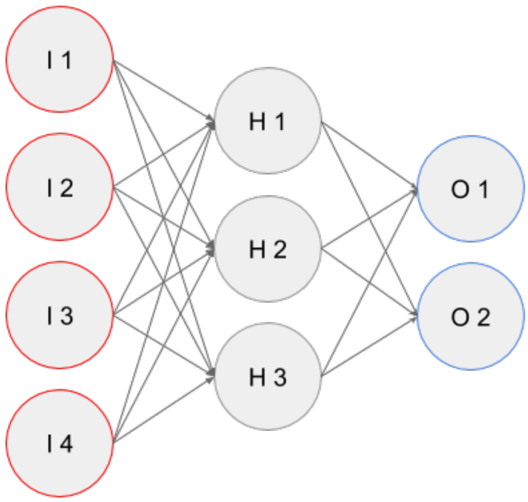


Figure 3.3: A three layer Feed-forward Neural Network

3.2.2 Convolutional Layer

Convolutional layer performs convolution operation on the input and pass the output to next layer. Figure 3.4 illustrates the convolution operation on a matrix. Convolution layer is usually followed by a pooling layer that reduces the dimension of the output of convolution layer. Convolutional layer enables automatic feature extraction from the input features.

3.2.3 Recurrent Layers

Recurrent layers gives the ability model sequence features. Generally it is used in identifying temporal features. However, it can also be used to model sequence of words in a sentence. Figure 3.5 illustrates a unrolled recurrent neural network architecture.

Repeating module in a RNN may have multiple layers depending on the type. Figure 3.6a and Figure 3.6b show a basic RNN module with only one layer and a LSTM module with four layers respectively. Although theoretically it is expected that a simple RNN to work for a long sequence, practically it is incapable of handling long sequences. However, LSTM [35] has the capability to handle longer sequences.

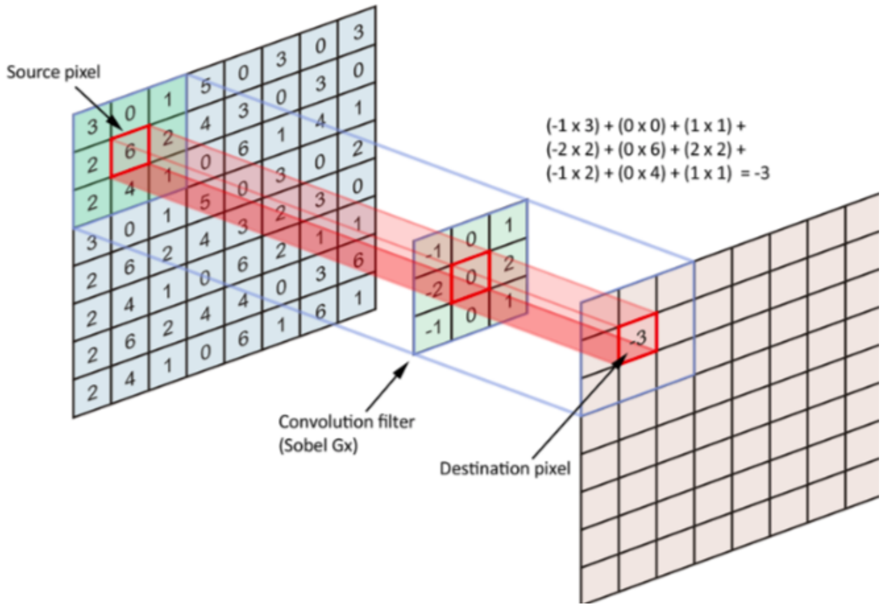


Figure 3.4: Convolution operation on a matrix
Source: https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050

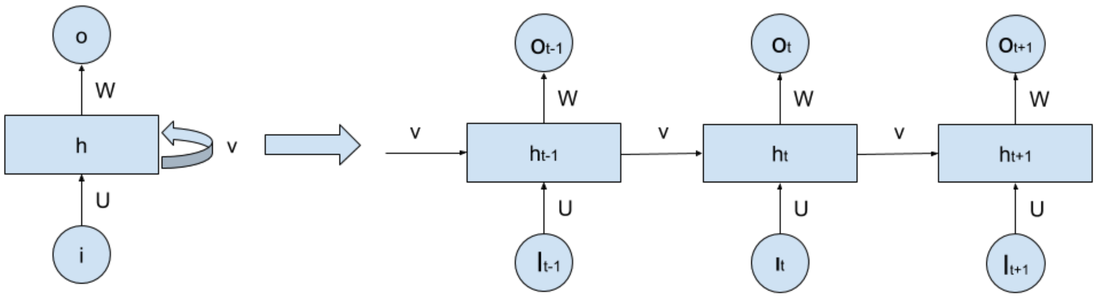
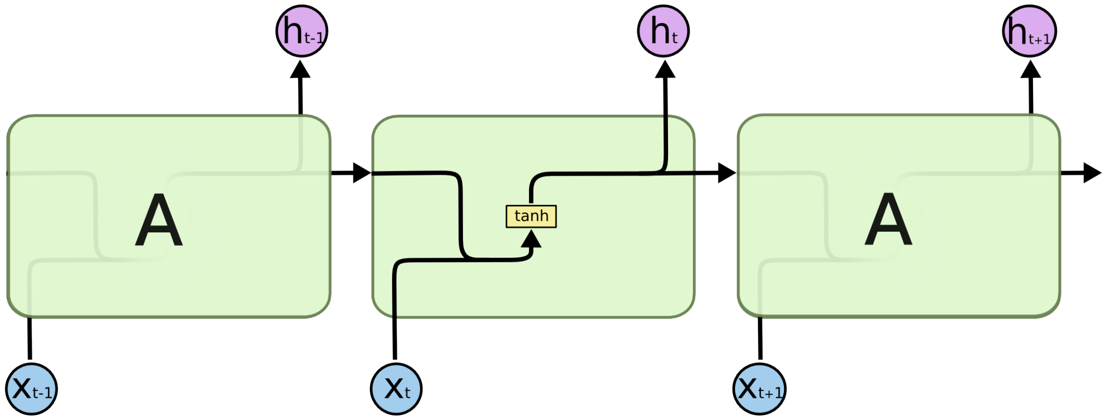


Figure 3.5: Unrolled RNN Structure

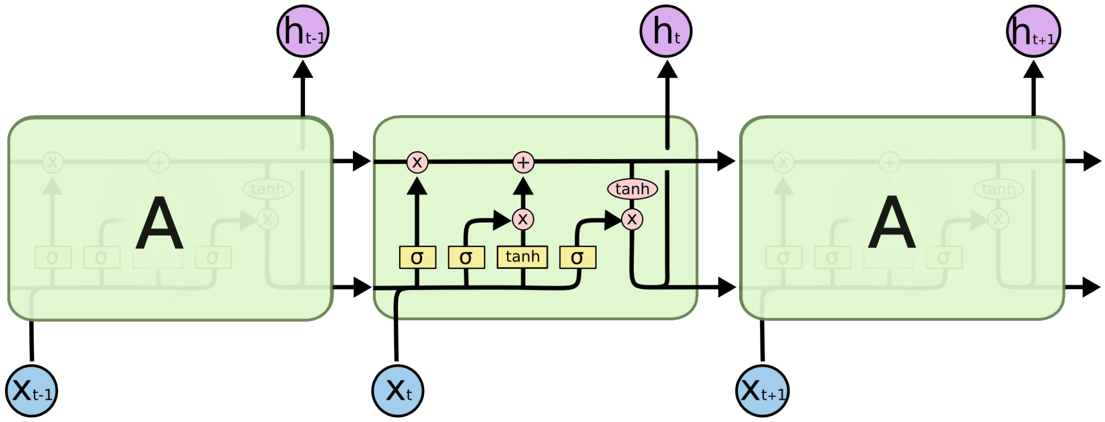
3.3 Transfer Learning

[36] defines transfer learning as follows: Given a source domain D_S and learning task T_S and a target domain D_T and learning task T_T, transfer learning aims to improve learning of the target predictive function f_T in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.

[37] describes four approaches to perform transfer learning and three different settings under which transfer learning can occur. A summary of different strategies for transfer learning is indicated in Figure 3.7. Table 3.1 shows the transfer learning approaches identified in [37].



(a) A simple RNN cell with single layer



(b) LSTM with four layers

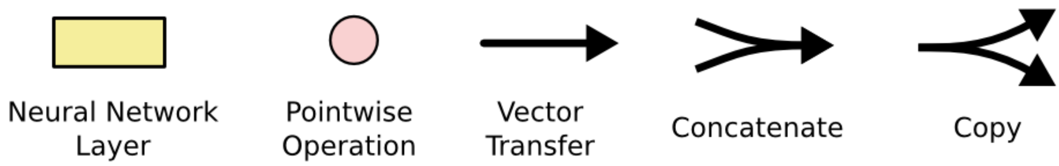


Figure 3.6: Common RNN module types
Source: https://colah.github.io/posts/2015-08-Understanding-LSTMs

Table 3.1: Transfer Learning Approaches

Transfer Learning Approach	**Description**
Instance Transfer	Re-weight and reuse labels in source domain to be used in target domain training.
Feature Representation Transfer	Identification of the best set of features that minimizes the differences between source and target domains.
Parameter Transfer	Find shared parameters to be used in target domain from a source domain model.
Relational Knowledge Transfer	Creating a relational knowledge mapping between source and target domain.

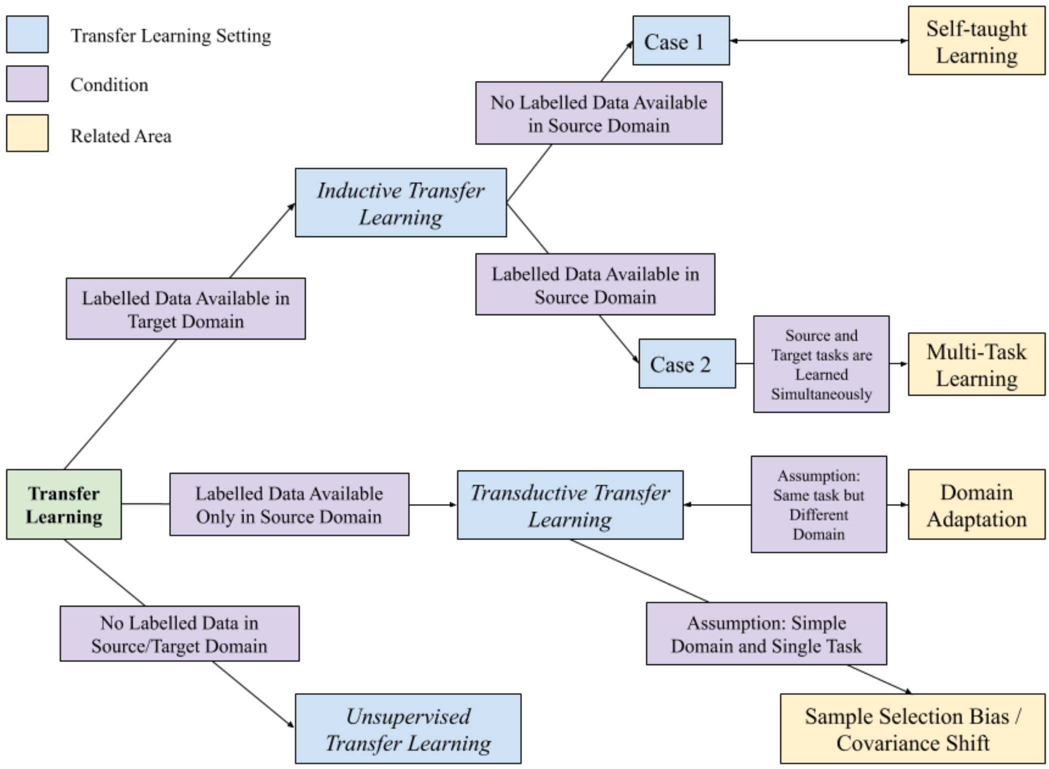


Figure 3.7: Different Settings of Transfer Learning

3.3.1 Transfer Learning with Fine-tuning

In order to transfer weights, we need to obtain a neural network model trained to solve a source task which has similarities to the domain of the target task. Then we can use the weights in the source model to initialize a similar neural network for the target task. Finally, we can train the neural network with the target domain data while fine-tuning the pre-trained weights that were transferred from the source model. Figure 3.8 indicates the high-level transfer learning process in neural network. This approach to transfer learning is identified under parameter transfer in [37].

Although this method will improve the performance of the target task using a similar model to source task, the computation requirement will be higher than training a separate model since generally the pretrained models have large number of parameters. Some examples for pretrained models that can be fine-tuned includes DeepMoji [21] and BERT [38].

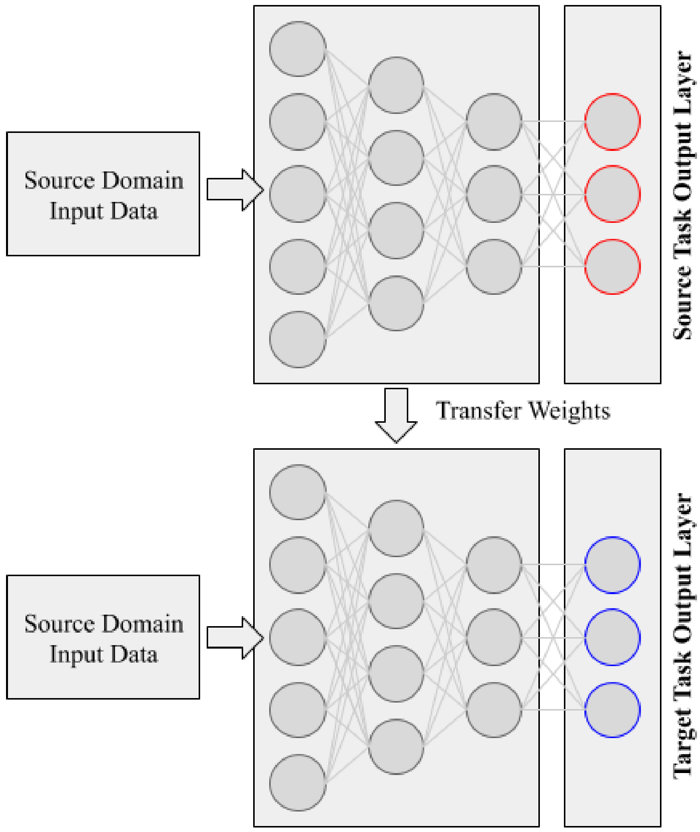


Figure 3.8: Transfer learning with Fine tuning

3.3.2 Transfer Learning with Fixed Model

Transfer learning with fixed model can be achieved by transferring only the features obtained from a pre-trained model. We can perform this by obtaining the output of a neural network layer (or even a single neuron). One advantage of using this approach over transferring the weights is that we can even use a simple neural network or traditional machine learning techniques such as SVM, Random Forest, and XG-Boost to train a model for the target task. This approach to transfer learning is classified under feature representation transfer in [37]. Figure 3.9 shows the main process.

Word Embedding models such as Word2vec [23], GLOVE [39], FastText [40] are commonly used to initialize word embedding layer of neural networks. These are the most common fixed model transfer features used in many neural network architectures in natural language processing tasks.

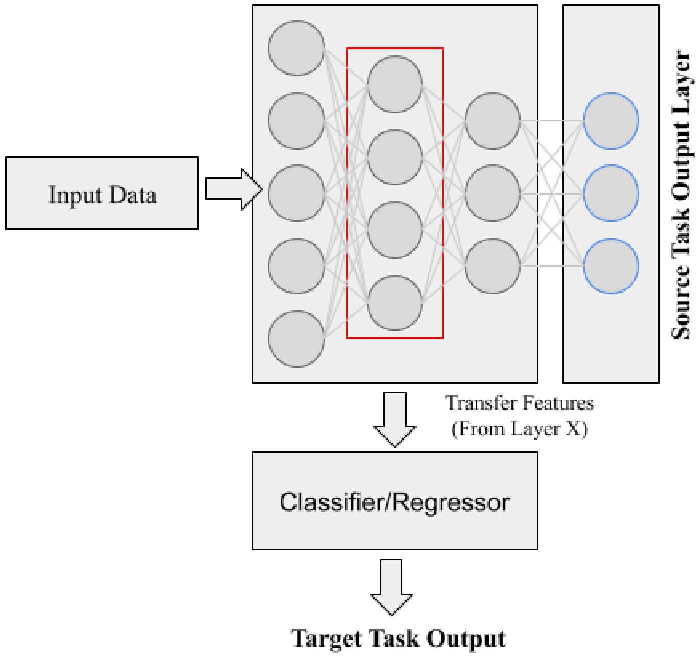


Figure 3.9: Transfer learning with Fixed Model

3.4 Word Embedding

Word embedding is a term used to identify the models that map word to a numerical vector representation. Multiple word embedding models has been introduced in the past. Among them Word2vec, Glove, and fastText takes a prominent place.

3.4.1 Word2vec

Word2vec is a neural word embedding model trained to predict either a word from the context (continuous bag of words) or the context from a word (continuous skip gram) in a sentence [23]. Word representation of a particular word is based on the hidden layer. It was identified that continuous skip gram models outperform continuous bag of words in multiple tasks and even has the ability to accurately represent infrequent words.

3.4.2 Glove

Glove [39] is an non neural based alternative to word2vec. Unlike word2vec Glove is obtained by optimizing word embeddings to represent individual words in the context. It optimizes the Function 3.1 where A, and B are words in Corpus C and $Count(A, B, C)$ is the count of words A and B appearing together (in some context window) in a corpus C.

$$vector(A) \cdot vector(B) = Count(A, B, C) \tag{3.1}$$

3.4.3 FastText

FastText [40] performs the model training similar to Word2vec. However, instead of using words in a sentence as the base units, it uses character n-grams. Therefore, fastText has the ability to predict the embedding for previously unseen words (words that were not available during training).

3.5 Cross Validation

Cross Validation (CV) is a technique used to evaluate the performance of classifiers/ regressor. Once the model has been trained using some dataset, the same dataset can't be used to evaluate the same model since the model can repeat what it has seen in the dataset without the capability to predict for new instances correctly. This is referred to as over-fitting in machine learning.

K-Flod is a basic form of cross validation. In K-Fold cross validation , we divide the dataset into k sections/folds and use each fold at a time as testing set while using other folds to train the model. Figure 3.10 illustrates this process.

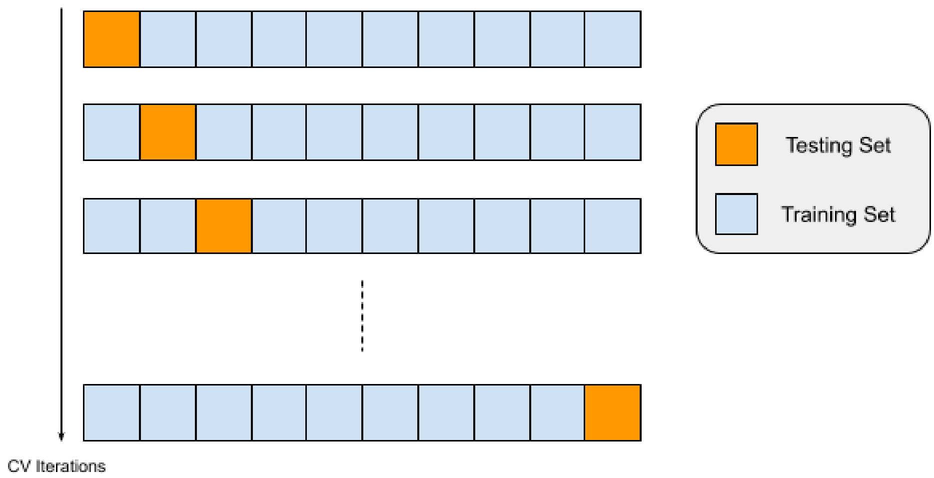


Figure 3.10: Cross validation iterations

Chapter 4

METHODOLOGY

This chapter explains the methodology used in this study to solve the emotion analysis problem. Figure 4.1 illustrates the approach used in this study. Since emotion analysis in general is broad topic, this chapter is divided in to three sections based on the achieved sub-objective. Section 4.1 describes the approaches used to extract the emotion category from tweets with implicit emotions. Section 4.2 provides the method used in predicting emotion intensity of tweets.

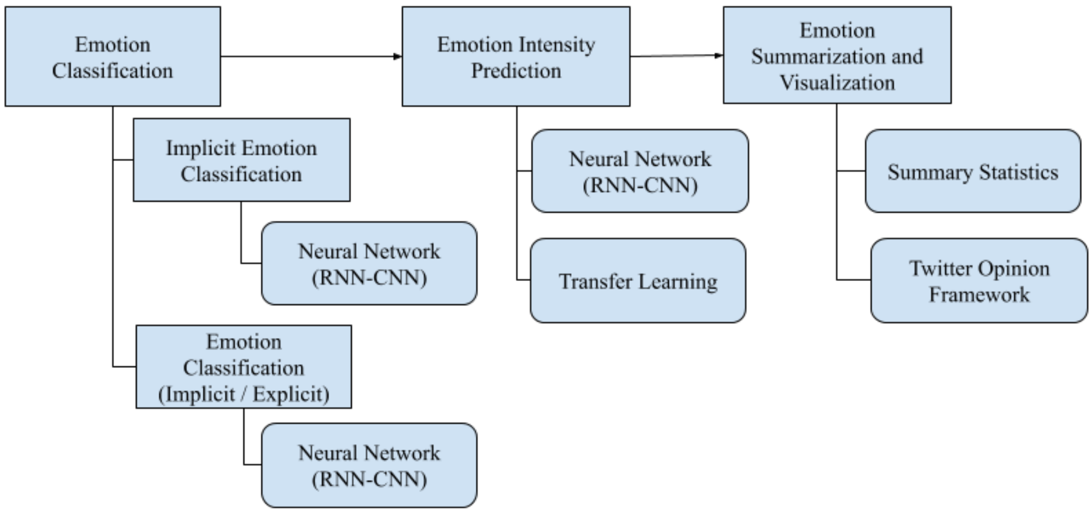


Figure 4.1: Approach used in this study for emotion analysis

4.1 Implicit Emotion Classification

Figure 4.2 depicts the overview of the implicit emotion classification system. It indicates a novel architecture based on concepts of both transfer learning and ensemble learning. Algorithm 15 shows the algorithm for training the implicit emotion classifier model. The algorithm takes the preferred classifier and training data as input.

Algorithm 1 Algorithm for Training Implicit Emotion Model

```
function TrainImplicitEmoClf(classifier, tweets, labels)
    tweet ← PreprocessTweet(tweet)
    X ← List()
    for all tweet ∈ tweets do
        for all feature ∈ features do
            x ← List()
            if feature typeOf Model then
                model ← getModel(feature)
                f ← model.predict(tweet)
                append f to x
            else
                f ← extractFeature(feature, tweet)
                append f to x
            append x to X
    return classifier.train(X, labels)
```

4.1.1 Tweet Pre-processor

The task of tweet pre-processor is to convert the tweet to a machine understandable format. This includes changes to the tweet text as well as the representation of tweet in vector space so that it could be understood by machine learning algorithms.

First, we replace Uniform Resource Identifiers (URIs) in tweet with a common format. This will alter all occurrences of various links as one type to the classifier. Mentions in tweets like "@someone" is replaced with a unique identifier to prevent bias against names. Additionally, new lines are marked with special tokens before passing to the tokenizer to prevent invalid tokenization.

Tokenization is performed to the tweets processed with above changes. This process splits the words in the tweet in the appropriate manner. We have used TweetTokenizer [1] to tokenize the text. Additionally, we evaluated our system using a dictionary based tokenizer.

[1] https://www.nltk.org/api/nltk.tokenize.html

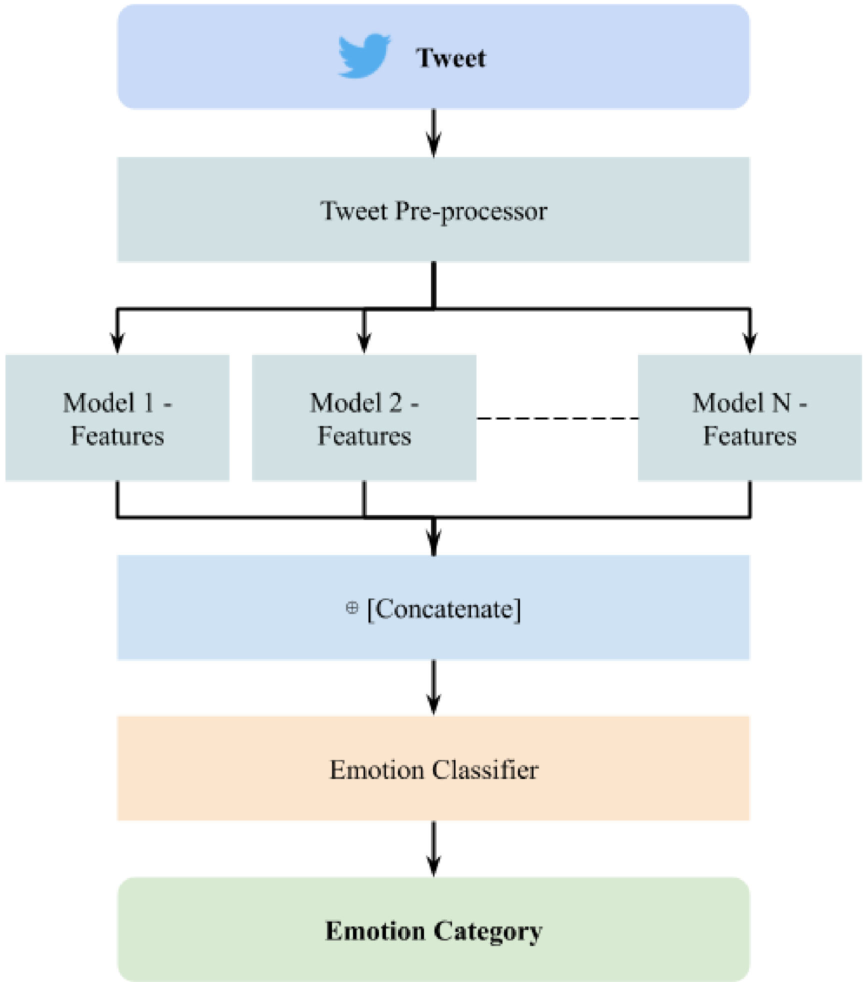


Figure 4.2: Overview of the implicit emotion classification architecture

4.1.2 Models

We experimented with deep learning classifiers FNN, CNN, RNN and their combinations to improve the performance of classification.

Figure 4.3 shows the high level architecture of the FNN we used in the experiments. The CNN model used in evaluations is architecturally similar to static multi-channel variant experimented by [41]. Furthermore, We used a LSTM based RNN to model a implicit emotion classifier. Additionally, we combined LSTM layer with convolutional layer to form LSTM-Conv network and Conv-LSTM network. In LSTM-Conv network the convolution layer follows LSTM layer while in Conv-LSTM convolution layer precedes LSTM layer. Figure 4.4 shows high level architecture of LSTM-Conv network.

Algorithm 2 Algorithm for Tweet Preprocessing

```
function PREPROCESSTWEET(tweet)
    x ← tweet
    for all url ∈ tweet do
        replace url in x by 'http://url.removed'
    for all mention ∈ tweet do
        replace mention in x by '@USERNAME'
    replace 'NewLineChar' in x by '__newline__'
    x ← tokenize(x)
    return x
```

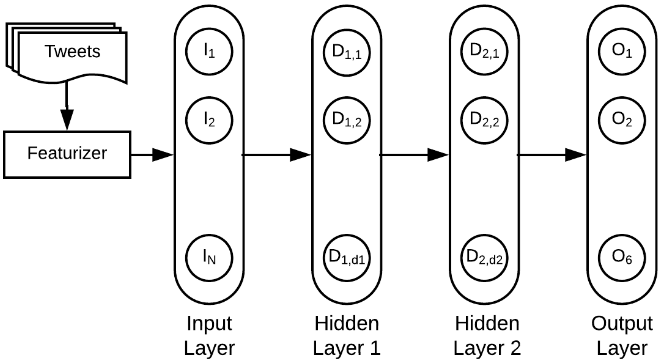


Figure 4.3: FNN used as as final implicit emotion classifier

4.1.3 Feature Extraction

A number of techniques have been developed to extract features for the classifier, some of which are trained on the dataset in order to create features explicitly. The most basic feature unit is the words. We used words to obtain the Word Vectors from multiple word embedding models trained on different corpses. Although our best performing system was based on word embeddings we developed and evaluated other features as well. In this section we will describe all the features that we have tried out.

Word Embedding: Table 4.1 summarizes all of the word embedding models we used in our implementation. It illustrates the word embedding techniques and the dataset it is trained on and its specific features as well. Additionally, it provides an identifier which we will be using to identify that word embedding in the next sections. Tweets can be represented as a word vector using the word2vec approach [23]. GW2V has been obtained by training Word2vec on part

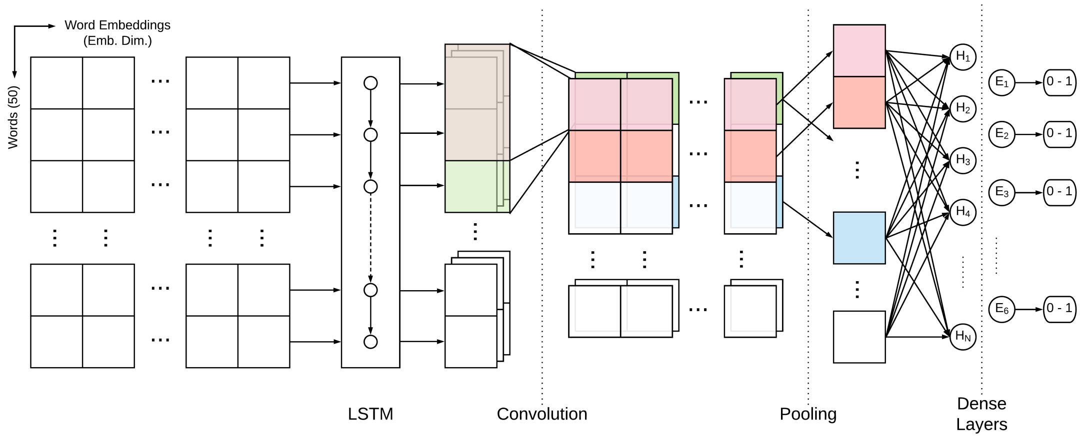


Figure 4.4: High-level LSTM-Conv Network Architecture

Abbr.	**Model**	**Corpus**	**Corpus Size**	**Dim**
TW2V	Word2Vec	Twitter	400M tweets	400
GW2V	Word2Vec	Google News	100B words	300
WFT	fastText	Wiki	16B tokens	300
WSFT	fastText	Wiki Subword	16B tokens	300
TGv	Glove	Twitter	2B tweets	200
E2V	Word2Vec	Twitter	1661 emoji	300

Table 4.1: Embedding Models used in Experiments

of Google News dataset[2]. Similarly, [42] has provided a word2vec model trained on twitter dataset (TW2V[3]). Furthermore, fastText [43] models are trained on trained on UMBC web-base corpus and statmt.org news dataset with and without sub-word information (WSFT and WFT)[4] [44]. Glove [39] embedding (TGv) has been trained on twitter corpus containing two billion tweets[5]. [22] has released emoji2vec (E2V)[6] a pre-trained embedding model for all Unicode emoji. Intended means of using E2V is as an extension to GW2V.

DeepMoji Features: DeepMoji [21] provides a pretrained model by training a Bi-LSTM attention network to predict emoji from tweet text. Here, we extract

[2]https://code.google.com/archive/p/word2vec/
[3]https://www.fredericgodin.com/software/
[4]https://fasttext.cc/docs/en/english-vectors.html
[5]https://nlp.stanford.edu/projects/glove/
[6]https://github.com/uclmr/emoji2vec

the features from attention layer of the model.

Context Embeddings: Context embedding is one of the common procedures for determining the embeddings for missing words based on the context. Context2vec can generate embedding for a target (missing) word provided the context [45].

Lexicon Features: [46] provides an extensive list of emotion and sentiment lexicons in AffectiveTweets [7] package.

Transfer Features: Features generated by training a neural classifier on the training dataset, obtained from the last layer (layer before the output later). Deep Learning models identified in Section 4.1.2 were used to extract features.

4.2 Emotion Intensity Prediction

Our emotion intensity prediction module is composed of several sub-modules. Figure 4.5 illustrates the overview of emotion intensity prediction module. This module is composed of Emotion Category Classifier Unit (ECCU) and Emotion Intensity Predictor Unit (EIPU) as feature generators for the final emotion intensity prediction module (EITL-Emotion Intensity Transfer Learning).

4.2.1 Tweet Pre-processor - Emotion Intensity Module

The tweet pre-processor used for predicting emotion intensity is different from the implicit emotion classification which was discussed in Section.

4.2.2 Models - Emotion Intensity

Emotion Category Classification Unit (ECCU):

Figure 4.6 depicts the architecture of ECCU sub-module. It is architecturally similar to LSTM-CNN module identified in Section 4.1.2. However, instead of using multiple channels we used single channel with fixed filter size.

[7]https://github.com/felipebravom/AffectiveTweets

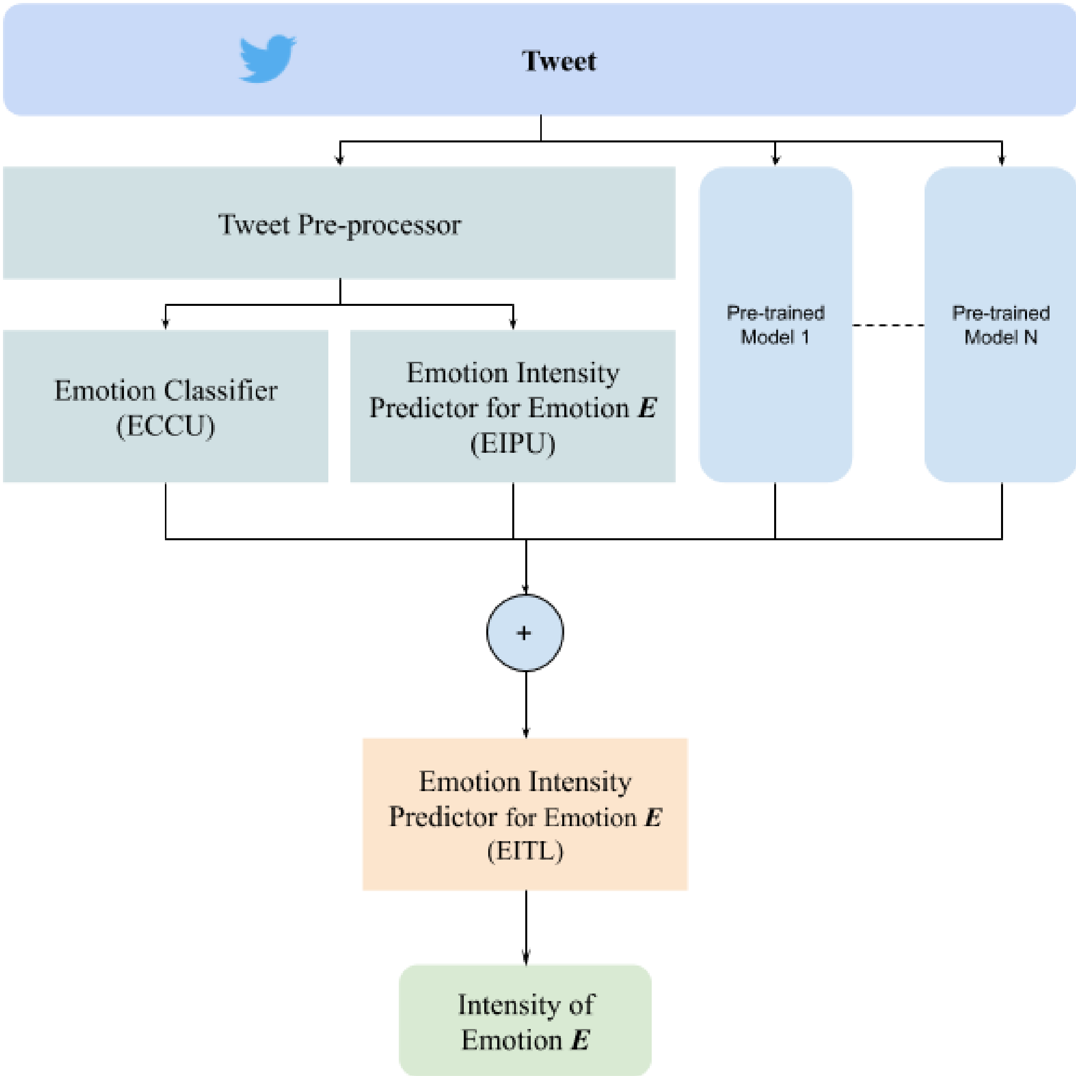


Figure 4.5: Overview of emotion intensity prediction architecture

Emotion Intensity Prediction Unit (EIPU):

The deep learning architecture used in EIPU is similar to that of ECCU. However, several changes were made to the architecture to make it compatible with emotion regression instead of classification. First modification is the replacement of the last layer of ECCU with a single Sigmoid neuron layer in place of multi-neuron layer that predicts the presence of each of emotion. Additionally, we created separate models that predict intensity of each emotion to support multiple emotions.

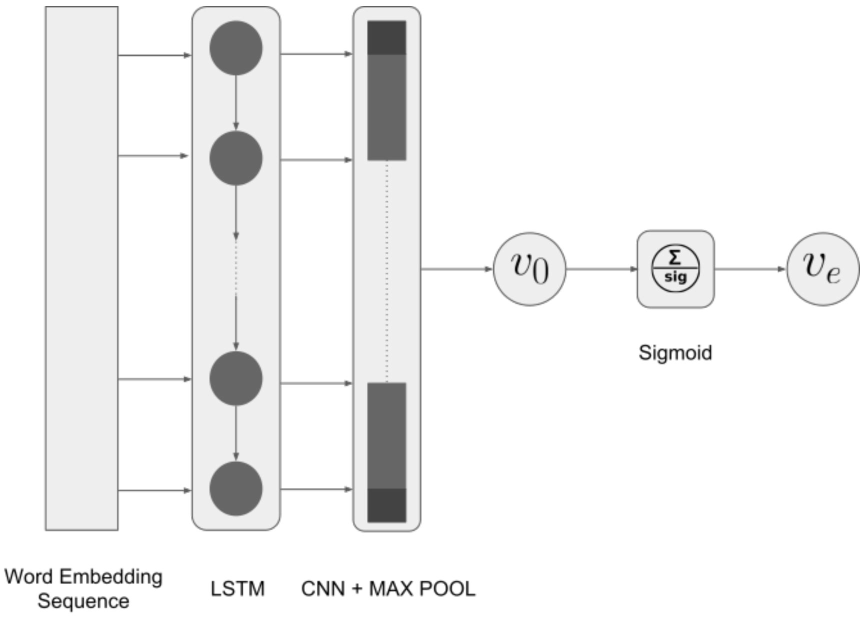


Figure 4.6: Recurrent-Convolutional Neural Network

Emotion Intensity Transfer Learning (EITL):

Figure 4.7 shows the high-level representation of EITL module. EITL module is composed of two parts:

1. Feature Extraction

    - **ECCU features**: Union of features from the output of max pooling layer (v_0) and Sigmoid layer (v_e) of ECCU model.
    - **EIPU features**: Union of features from the output of max pooling layer (v_0) and Sigmoid layer (v_e) of EIPU model.
    - **DeepMoji features**: Union of features from attention layer and soft-max layer of pre-trained DeepMoji model [21] [8].
    - **Sentiment Neuron**: Features from pre-trained unsupervised sentiment neuron model [9] [11].

2. Regression with XG-Boost

    - We use XGBoost [47] regressor as the target predictive function

[8] https://github.com/bfelbo/DeepMoji
[9] https://github.com/openai/generating-reviews-discovering-sentiment

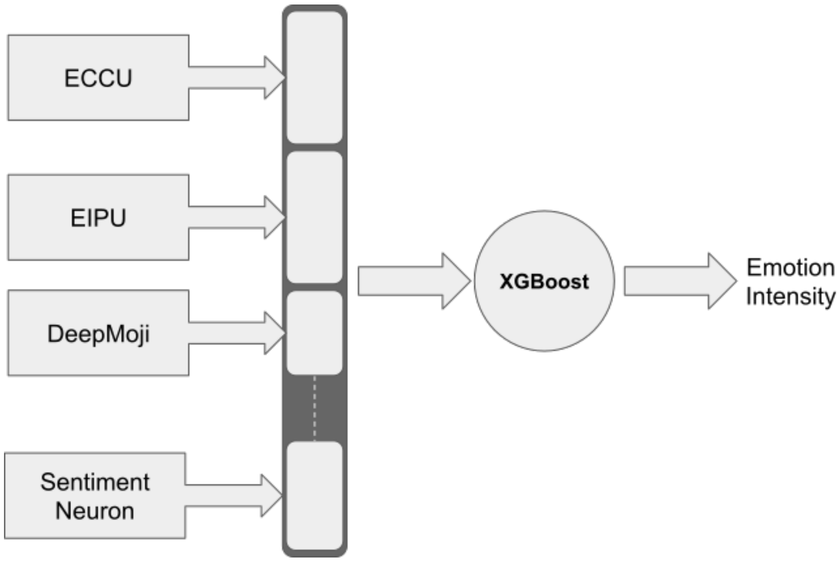


Figure 4.7: Emotion Intensity Regression - EITL Module

Chapter 5

EVALUATION

This chapter explains the evaluation process and the results of models identified in Section 4. This chapter is divided in to two sections. Section 5.1 describes the datasets and evaluation procedure of implicit emotion classification module. Section 5.2 provide the datasets and evaluation of emotion intensity prediction module and its sub-modules.

5.1 Implicit Emotion Classification

In this section, we show evaluations of the models identified in Section 4.2.

5.1.1 Dataset

We used IEST: WASSA-2018 Implicit Emotions Shared Task [1] dataset to train and evaluate the models [24]. This dataset is composed of tweets obtained from the web containing the expression '`<Emotion-Word> (that|because|when)`', where '<Emotion-Word>' indicates a word identifying an emotion from Ekman's six basic emotions [12]. Table 5.1 shows the basic emotions and words used to identify the emotions.

The `<Emotion-Word>` was then removed from the extracted tweets and replaced with a marker for the position of the emotion word. Each tweet was them labelled with the emotion category related with the removed emotion word. Some examples from the dataset are:

- "It's TRIGGER_WORD when you feel like you are invisible to others." (Emotion: Sad)
- "I'm kinda TRIGGER_WORD that I have to work on Father's Day." (Emotion: angry)

[1] http://implicitemotions.wassa2018.com/

Table 5.1: Emotion words used when collecting Tweets

Emotion	**Abbr.**	**Synonyms**
Anger	A	angry, furious
Fear	F	afraid, frightened, scared, fearful
Disgust	D	disgusted, disgusting
Joy	J	cheerful, happy, joyful
Sadness	Sa	sad, depressed, sorrowful
Surprise	Su	surprising, surprised, astonished, shocked, startled, astounded, stunned

Table 5.2: Distribution of the Implicit Emotion Dataset

Emotion	**Train**	**Dev**	**Test**
Anger	25562	1600	4794
Disgust	25558	1597	4794
Fear	25575	1598	4791
Joy	27958	1736	5246
Sadness	23165	1460	4340
Surprise	25565	1600	4792
Sum	153383	9591	28757

Table 5.2 shows the distribution of labelled tweets in the dataset.

5.1.2 Experimental Setup

This section holds the details on experimental setup of the best model from the models identified in the Section 4.1.

Table 5.3 shows the optimized hyper-parameters for LSTM-CNN model training. A combination of manual and tool based hyper-parameter optimization was

Table 5.3: Network Parameters for LSTM-CNN

Section	Parameter	Value
LSTM	Num. of units	250
CNN	Num. of filters	350
	Kernel Sizes	2, 3, 5
Pooling	Method	Max
Dense Layer	Num. of units	50
	Activation	ReLU
Output Layer	Num. of Units	6
	Activation	Softmax

Table 5.4: Network parameters for FNN

Section	Parameter	Value
Hidden Layer 1	Num. of Units	50
	Activation	ReLU
Hidden Layer 2	Num. of Units	25
	Activation	ReLU

performed. We used Tree of Parzen Estimators (TPE) [48] to optimize parameters of LSTM-CNN model. However, a long time was required to optimize hyper-parameters using this method due to large dataset size and complexity of the neural network. Therefore, we adopted a manual hyper-parameter optimization method in the later part of the study. Hyper-parameter optimization was based on the results on the development (Dev) subset of the dataset.

We used categorical cross-entropy between the ground truth and predicted labels to optimize the neural network parameters. Parameter optimization is done by back propagation with mini-batch gradient descent. We used batch size of 256 and trained the network for five (5) epochs. To prevent overfitting, we used a dropout layer with rate 0.2 before the last dense layer. To improve the performance of the optimization algorithm we used Adam optimization algorithm [49].

Table 5.4 illustrates the network-configuration for the FNN. We used training parameters similar to that of LSTM-CNN when training the FNN. However, we had to apply dropout layers with dropout rate 0.5 after each dense layer to prevent overfitting. We extracted the features from LSTM-CNN models trained to predict emotions with different word embeddings. The best set of features were obtained when the models were initialized with embeddings: TW2V, GW2V + E2V and WFT.

5.1.3 Results

We measure the impact of using different word embeddings to train the LSTM-CNN models in our first set of experiments. Table 5.5 shows the results of our experiments with LSTM-CNN. Although we trained the models with only train data when evaluating with the trial set, we combined the train set and trial set

Table 5.5: Evaluation of LSTM-CNN for different word embeddings

Abbr.	**Features**	**Trial Set (%)**			**Test Set (%)**		
		Macro Precision	**Macro Recall**	**Macro F1**	**Macro Precision**	**Macro Recall**	**Macro F1**
M_{TW2V}	TW2V	**65.9**	**65.5**	**65.5**	**67.1**	**67.0**	**67.0**
M_{E2V}	GW2V+E2V	63.7	63.6	63.6	65.6	65.1	65.2
M_{GW2V}	GW2V	64.4	62.6	62.9	65.4	63.7	63.8
M_{WFT}	WFT	65.3	64.1	64.3	65.5	65.1	65.2
M_{WSFT}	WSFT	62.5	62.0	62.0	63.9	62.2	62.5
M_{TGv}	TGv	63.4	63.2	63.2	63.9	63.9	63.9
Baseline (MaxEnt Classifier)		60.1	60.1	60.1	-	-	59.8

Table 5.6: Results of FNN for best feature combinations

Features	**Macro Precision (%)**	**Macro Recall (%)**	**Macro F1 (%)**
$F(M_{TW2V}) \oplus F(M_{E2V})$	68.0	67.8	67.8
$F(M_{TW2V}) \oplus F(M_{WFT})$	67.9	67.8	67.8
$F(M_{E2V}) \oplus F(M_{WFT})$	67.1	66.7	66.8
$F(M_{E2V}) \oplus F(M_{TW2V}) \oplus F(M_{WFT})$	68.3	68.1	68.1
Baseline	-	-	59.8
Amobee [50]	-	-	**71.5**

to obtain models for the test set evaluations.

Table 5.6 provides results of the FNN models trained using features extracted from LSTM-CNN models. '$\oplus$' represents vector concatenation operation and $f(M, tweet)$ is a function that maps a *tweet* to a set of features using model M by extracting features from the last dense layer of the model. We have removed *tweet* argument from $f(M, tweet)$ in Table 5.6 when representing. We performed the evaluations with the three best performing LSTM-CNN models: M_{TW2V}, M_{WFT} and M_{E2V}. We excluded M_{GW2V} for this analysis because M_{E2V} includes word vectors in M_{GW2V}.

5.1.4 Discussion

We can observe similar performance changes across the trial-dataset and test-dataset for the different models that we have experimented from Table 5.5. Additionally, we can observe a similar or improved performance on test-dataset compared to the results on the trial-dataset. Word2vec trained on Twitter corpus provided the best performance because it contains word vectors for in-domain words. Moreover, we observe an improvement when we expand the vocabulary of M_{GW2V} with Emoji2Vec. Therefore, we can presume the importance of emoji for the implicit emotion classification models. Furthermore, we can infer that the sub-word information provided by the embeddings are not essential for the implicit emotion classification by observing the performance of M_{WFT} and M_{WSFT}. We can observe that the models in Table 5.5 outperforms the baseline model demonstrating the capacity to model tweets with implicit emotions. The baseline model is a maximum-entropy classifier with L2 regularization. The baseline classifier was trained with boolean features extracted from unigrams and bigrams.

The performance of FNN models identified in Table 5.6 is better than individual model shown in Table 5.5. Table 5.6 reveals that the combination of features from M_{TW2V} and M_{E2V} provides a better performance. The system performs best when we combined th features from M_{TW2V}, M_{E2V} and M_{WFT} models.

5.2 Emotion Intensity Prediction

This section presents the datasets, experimental setup, results and analysis for the emotion intensity prediction models.

5.2.1 Dataset

Emotion intensity prediction was based on two main sub-modules: ECCU and EIPU. ECCU was trained and evaluated on dataset from SemEval-2018 Task 1: Affect in Tweets for Emotion Classification (E-c) task[16]. EIPU was trained and evaluated on the SemEval-2018 Task 1: Affect in Tweets for Emotion Regression (EI-reg) dataset.

Table 5.7: The number of tweets in the SemEval-2018 Affect in Tweets Dataset

Task	**Train**	**Dev**	**Test**	**Total**
E-c	6,838	886	3,259	10,983
EI-reg				
anger	1,701	388	1,002	3,091
fear	2,252	389	986	3,627
joy	1,616	290	1,105	3,011
sadness	1,533	397	975	2,905

Each tweet in emotion classification dataset is annotated for presence/absence of 11 emotions. List of annotated emotions are: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust. Table 5.7 shows the statistics for the emotion classification (E-c) and Emotion intensity regression datasets (EI-reg).

5.2.2 Experimental Setup

We identified set of best parameters for ECCU, EIPU and EITL models by evaluating the model with the development (Dev) set of the datasets and cross validation. Table 5.8 indicates the hyper-parameters and training parameters used in training the ECCU, EIPU models. Table 5.9 shows the training parameters and the features used in the EITL model.

The embedding layer was initialized with Twitter specific word2vec published in [17]. We maintain the original word embedding by preventing the training algorithm from fine-tuning the word embedding layer. We used categorical cross-entropy loss and mean squared error as loss function when training ECCU and EIPU respectively. Adam optimizer is used since it can generate better results fast. The hyper-parameters for training the proposed network was based on results on validation dataset provided in SemEval Task [16].

5.2.3 Results

Table 5.10 indicates the performance score for classification model using accuracy, micro F1 and macro F1 scores. We clearly see that ECCU outperforms the

Table 5.8: Model and training hyper-parameters for ECCU and EIPU models. *Number of epochs for Anger emotion intensity model is 40 while for all other emotions we used 15.

Parameter	**ECCU**	**EIPU**
LSTM		
Units	128	64
Dropout Rate	0.5	0.8
Convolutional Layer		
Filters	128	64
Kernel size	2	2
Padding	Same	Same
Activation	ReLU	ReLU
Dropout Layer		
Rate	0.5	0.8
Last (Dense) Layer		
Activation	Sigmoid	Sigmoid
Training		
Number of Epochs	10	15/40*
Batch Size	8	8

benchmark system and a random baseline. Here we obtained random baseline results by a system that randomly guesses the prediction.

In Table 5.11 we compare our models against existing systems and three strong baselines. We obtained the baseline results from NTUA-SLP [17]. The first baseline is the unigram Bag-of-Word (BoW) model with TF-IDF weighting. Second baseline is the Neural BoW (NBoW) model, constructed by averaging the word2vec embedding of words in a Tweet. Last baseline is similar to NBoW

Table 5.9: Parameters used for training EITL model. C1 indicates parameters for Anger, Joy, Sadness emotion intensity models. C2 indicates training parameters for Fear emotion intensity prediction model.

Parameter/Feature	**C1**	**C2**
Features		
DeepMoji features	✓	✓
Sentiment Neuron	✓	✓
ECCU features	✓	✓
EIPU features	✓	✗
Max Depth	2	5
Learning Rate	0.01	0.01
# of Estimators	400	300

Table 5.10: Performance scores for ECCU compared with the benchmark systems. The marker † indicates the benchmark [17].

Model	**Accuracy (Jaccard)**	**F1 (Micro)**	**F1 (Macro)**
ECCU	**58.63**	**71.92**	**52.8**
NTUA-SLP†	57.88	70.1	**52.8**
Random Baseline	18.5	30.7	28.5

Table 5.11: Performance scores of emotion intensity prediction models. The marker † indicates the benchmark [51] and * indicates results obtained in [17]

Model	**Pearson Correlation**				**Average**
	Anger	**Fear**	**Joy**	**Sadness**	
EIPU	76.45%	67.08%	72.10%	68.95%	70.83%
EITL	82.16%	**78.67%**	78.42%	**79.99%**	79.81%
SeerNet†	**82.70%**	77.90%	**79.20%**	79.80%	**79.90%**
NTUA-SLP*	78.20%	75.80%	77.10%	79.80%	77.70%
BoW	52.49%	52.27%	57.16%	47.21%	52.28%
NBoW	65.39%	63.18%	63.55%	63.05%	63.79%
NBoW+A	65.60%	63.59%	63.84%	63.41%	64.11%

except it has extra 10-dimensions in the embedding with affective information (NBoW+A). Aforementioned features are used as inputs to an SVM with C=0.6 to obtain the baselines.

5.2.4 Discussion

We observe that our neural model (EIPU) outperforms the baselines with substantial performance improvement. Moreover, we see that our proposed transfer learning model outperforms the existing state of the art models for two emotions: fear and sadness while maintaining competitive results over other emotions. Additionally, we clearly exceed the NTUA-SLP [17], the second best system at SemEval 2018 EI-reg sub-task of Emotion in Tweet task. However, EIPU did not perform well with respect to the transfer learning based models in Table 5.11. This behaviour can be attributed to the extra information provided through transfer learning.

Table 5.12 shows the word-level importance of the emotion intensity prediction of EIPU model. Here, columns represented by letters E, A and P represents

Table 5.12: Examples for word level importance heat-map visualizations.

E	Tweet	A	P
A	[user] # crap sorry but doesn ' t work	0.563	0.556
A	[user] i will second that # arsehole	0.697	0.690
F	johnny does not seem like the suicidal type # suspicion # tcm party	0.620	0.610
F	so i now have [number] pairs of shit arsehole neighbours ; any advice ? # noisy # no respect # big building # awful # killing in the name of	0.625	0.576
J	there ' s no pine in it , there ' s no apple in it , let us call it pineapple # funny	0.547	0.537
J	[user] glad it went well 😄	0.656	0.647
S	is it okay to think you are going to die alone ? # sadness	0.731	0.723
S	what a tiring day 😟	0.636	0.6277
A	# lt a mom let her kid pay for the bus . the kid dropped the coins . i am boiling in rage . my face may be straight bu my eyes say [allcaps] imgonnakillya [/allcaps]	0.848	0.505
A	nothing joyful has happened to me . my grief and hatred just keep increasing !	0.813	0.425
F	[user] the possibilities terrify me	0.911	0.562
F	yeah im more afraid	0.913	0.578
J	i am in such a good place right now , so happy . [repeated] 💛 i just wanted to share that # positive vibes # optimism # following my dream	0.955	0.668
J	i love being outside right before sunrise / after sunset because the sky is so bright blue and it brings me peace	0.828	0.560
S	i miss my boys so much 😭 rest well babies [user] 😚	0.696	0.360
S	bout to cry 😭 i just wanna sleep i hate this job 😩	0.786	0.454

emotion category, true emotion intensity and predicted emotional intensity respectively. Characters A, F, J and S in column E corresponds to emotions Anger, Fear, Joy and Sadness. The predicted emotion intensity of tweets above the double-line separator is closer to the actual value while the difference between the actual and predicted is significantly higher for the tweets below that separator.

Chapter 6

Twitter Emotion Analysis Platform

This section will layout the architecture and details of the emotion summarization and visualization platform. Figure 6.1 shows the high-level architecture of web based solution to emotion summarization and visualization. Here, we have a end to end platform for emotion visualization from collection of tweets using Tweet API using a keyword to summarization and emotion intensity distribution visualization through box-plots and bar charts.

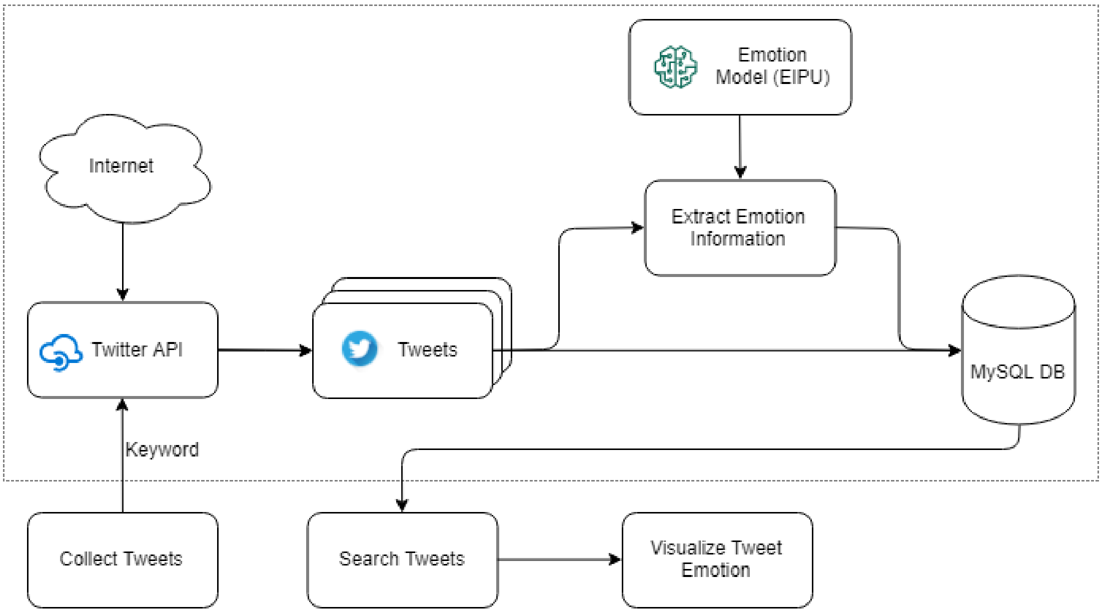


Figure 6.1: High-level Architecture of Emotion Visualization/Summarization Architecture

Emotion Information Extraction unit in Figure 6.1 performs language identification to check whether it complies with the language of the model (English). If they match, the emotion intensity of each emotion is calculated and stored in the database.

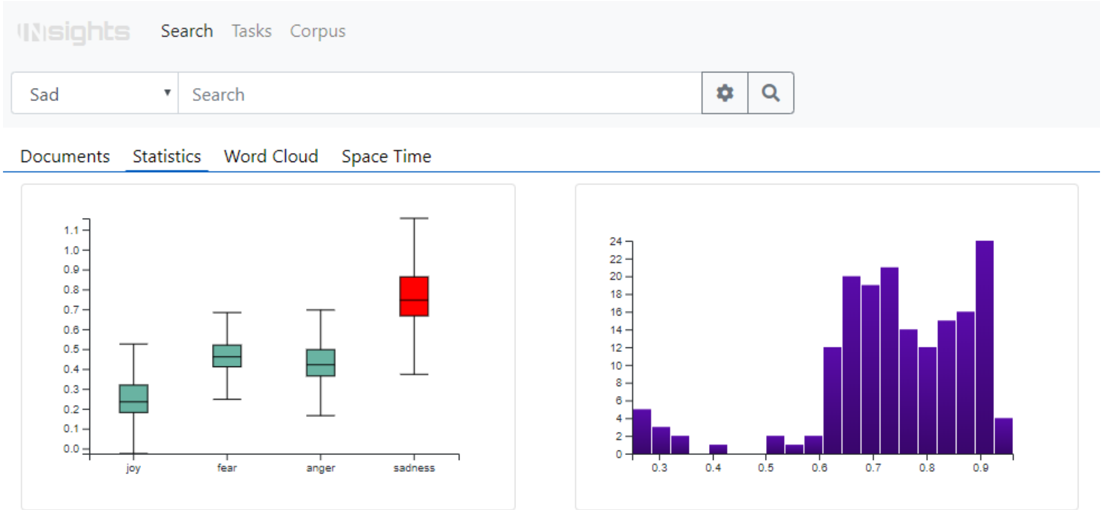


Figure 6.2: Emotion visualization and summarization platform

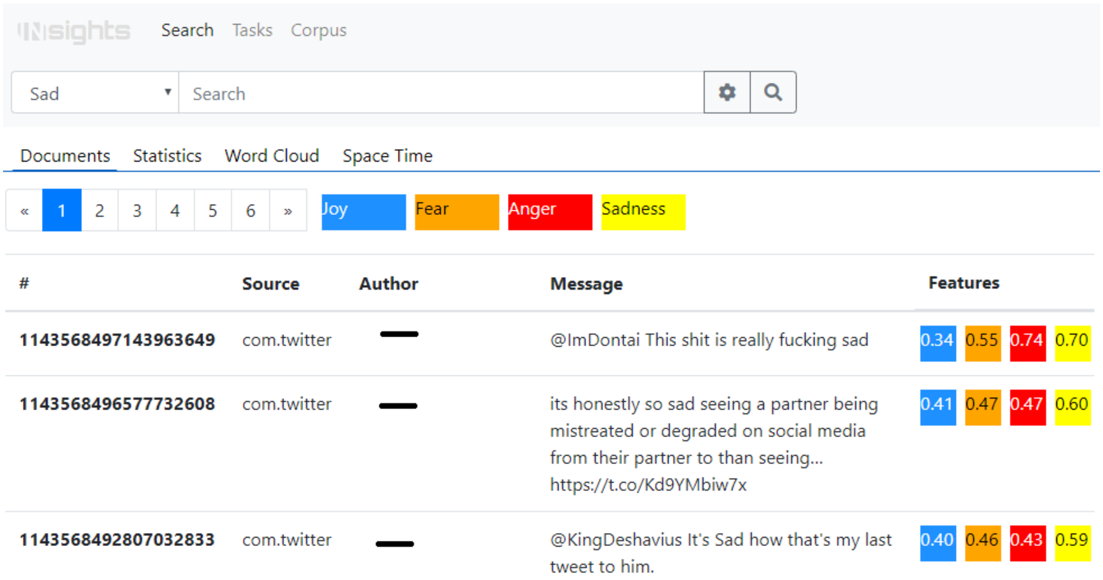


Figure 6.3: Individual emotion intensity visualization with the visualization platform

Figure 6.2 shows our emotion visualization platform at work. Here user has searched for the emotion summary for all tweets in "Sad" corpus that was initialized by obtaining tweets containing "sad" using Twitter corpus. Additionally, Figure 6.3 illustrates a screen-shot of this platform showing individual emotion intensity of tweets in a corpus.

Chapter 7

CONCLUSION

In this research, we were able to analyze the Tweet emotion extraction process comprehensively. As a result, we were able to create an end to end emotion analysis system that identifies emotions and associated emotion intensities of a tweet.

We recognized the identification of implicit emotions as a central challenge in the emotion analysis process. In the initial phase of this study, we introduce a novel architecture for implicit emotion classification that enables reuse of the same neural network architecture for extraction of different features. We combined two popular deep learning architectures, CNN and RNN in a single neural network to create the basic neural network. Similarly, we create multiple copies of this network and train them using the same dataset while changing the underlying word embedding model. The results show that individual neural network models outperform the maximum entropy baseline classifier trained with unigram and bi-gram features. These results suggest that our neural networks were able to identify better features for implicit emotion classification. Then, we used transfer learning to transfer features identified when training individual neural network models and use a second neural network to make the final classification. We use a Feed-forward Neural Network (FNN) as the final classifier since we do not need a complex feature extraction process. The improved results of FNN reveal that differences between initial word embedding models used in training basic neural networks contributed in extracting some distinct features.

We studied emotion intensity prediction in the second phase of our research. We used a neural network that is architecturally similar to the basic neural network we used for implicit emotion classification. Moreover, we trained a similar neural network for classifying emotions in general. Additionally, we identified two popular pre-trained models used in sentiment classification, and emoji classification for feature transferring. We trained secondary XG-Boost regressor with combined features from emotion intensity predictor, emotion classifier, sentiment classifier, and emoji classifier to obtain the final emotion intensity prediction. The evaluation shows that our transfer learning approach outperforms existing similar systems on two emotions. Furthermore, our emotion classifier outperforms existing systems for emotion classification.

Finally, we created an end to end emotion analysis platform using the models that we have created. Keywords serve as the input to the system. The system will initiate a task that requests Tweets containing the input keyword using the Twitter API. We can query for Tweets and obtain the intensity of emotions for each resulting Tweet and the summary of emotion intensity for each emotion.

7.1 Future Work

In the future, we will try to extend the emotion analysis process for conversations. Currently, we target the analysis of tweets individually. However, related tweets such as replies can provide additional information to predict emotions accurately.

Our system is capable of predicting emotion intensity of four emotions: Anger, Fear, Joy, and Sadness. However, there are eight primary emotions identified in [13]. Therefore, we can widen the scope of our system to analyze other emotions by expanding the dataset.

Furthermore, we would like to adopt this platform to analyze low resource languages such as Sinhalese and Tamil. Our current work on emotion analysis for such languages is at the preliminary stages.

References

[1] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 70–74, 2007.

[2] Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R Zaiane. Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR)*, 50(2):25, 2017.

[3] Yasas Senarath and Uthayasanker Thayasivam. Datasearch at iest 2018: Multiple word embedding based models for implicit emotion classification of tweets with deep learning. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 211–216, 2018.

[4] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[6] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 451–463, 2015.

[7] Svetlana Kiritchenko, Saif Mohammad, and Mohammad Salameh. Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the 10th international workshop on semantic evaluation (SEMEVAL-2016)*, pages 42–51, 2016.

[8] Hugo Liu, Henry Lieberman, and Ted Selker. A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125–132. ACM, 2003.

[9] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing 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)*, pages 1555–1565, 2014.

[10] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 49–54, 2014.

[11] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

[12] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[13] Robert Plutchik. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350, 2001.

[14] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 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*, pages 579–586. Association for Computational Linguistics, 2005.

[15] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[16] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, 2018.

[17] Christos Baziotis, Athanasiou Nikolaos, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 245–255, 2018.

[18] Sara Rosenthal, Noura Farra, and Preslav Nakov. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518, 2017.

[19] Hardik Meisheri and Lipika Dey. Tcs research at semeval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299, 2018.

[20] Ji Ho Park, Peng Xu, and Pascale Fung. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and# hashtags. *arXiv preprint arXiv:1804.08280*, 2018.

[21] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *2017 Conference on Empirical Methods in Natural Language ProcessingConference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017.

[22] Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. emoji2vec: Learning emoji representations from their de-

scription. In *Conference on Empirical Methods in Natural Language Processing*, page 48, 2016.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[24] Roman Klinger, Orphee De Clercq, Saif Mohammad, and Alexandra Balahur. Iest: Wassa-2018 implicit emotions shared task. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 31–42, 2018.

[25] Phil Katz, Matthew Singleton, and Richard Wicentowski. Swat-mp: the semeval-2007 systems for task 5 and task 14. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 308–313. Association for Computational Linguistics, 2007.

[26] François-Régis Chaumartin. Upar7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 422–425. Association for Computational Linguistics, 2007.

[27] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Compositionality principle in recognition of fine-grained emotions from text. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[28] Saif M Mohammad and Felipe Bravo-Marquez. Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*, 2017.

[29] Jordan J Louviere and George G Woodworth. Best-worst scaling: A model for the largest difference judgments. *University of Alberta: Working Paper*, 1991.

[30] Pranav Goel, Devang Kulshreshtha, Prayas Jain, and Kaushal Kumar Shukla. Prayas at emoint 2017: an ensemble of deep neural architectures for

emotion intensity prediction in tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 58–65, 2017.

[31] Maximilian Köper, Evgeny Kim, and Roman Klinger. Ims at emoint-2017: emotion intensity prediction with affective norms, automatically extended resources and deep learning. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–57, 2017.

[32] Venkatesh Duppada and Sushant Hiray. Seernet at emoint-2017: Tweet emotion intensity estimator. *arXiv preprint arXiv:1708.06185*, 2017.

[33] Vineet John and Olga Vechtomova. Uwat-emote at emoint-2017: emotion intensity detection using affect clues, sentiment polarity and word embeddings. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 249–254, 2017.

[34] Md S Akhtar, Palaash Sawant, Asif Ekbal, JD Pawar, and Pushpak Bhattacharyya. Iitp at emoint-2017: Measuring intensity of emotions using sentence embeddings and optimized features. Association for Computational Linguistics, 2017.

[35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[36] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[37] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[40] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, April 2017.

[41] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[42] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, 2015.

[43] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[44] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[45] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, 2016.

[46] Saif Mohammad and Felipe Bravo-Marquez. Emotion intensities in tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational*

*Semantics, *SEM @ACM 2017, Vancouver, Canada, August 3-4, 2017*, pages 65–77, 2017.

[47] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[48] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[50] Alon Rozental, Daniel Fleischer, and Zohar Kelrich. Amobee at iest 2018: Transfer learning from language models. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 43–49, 2018.

[51] Venkatesh Duppada, Royal Jain, and Sushant Hiray. Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 18–23, 2018.