INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación

TESIS

Social Media Emotion Detection

Que para obtener el grado de:

Maestría en Ciencias de la Computación

PRESENTA:

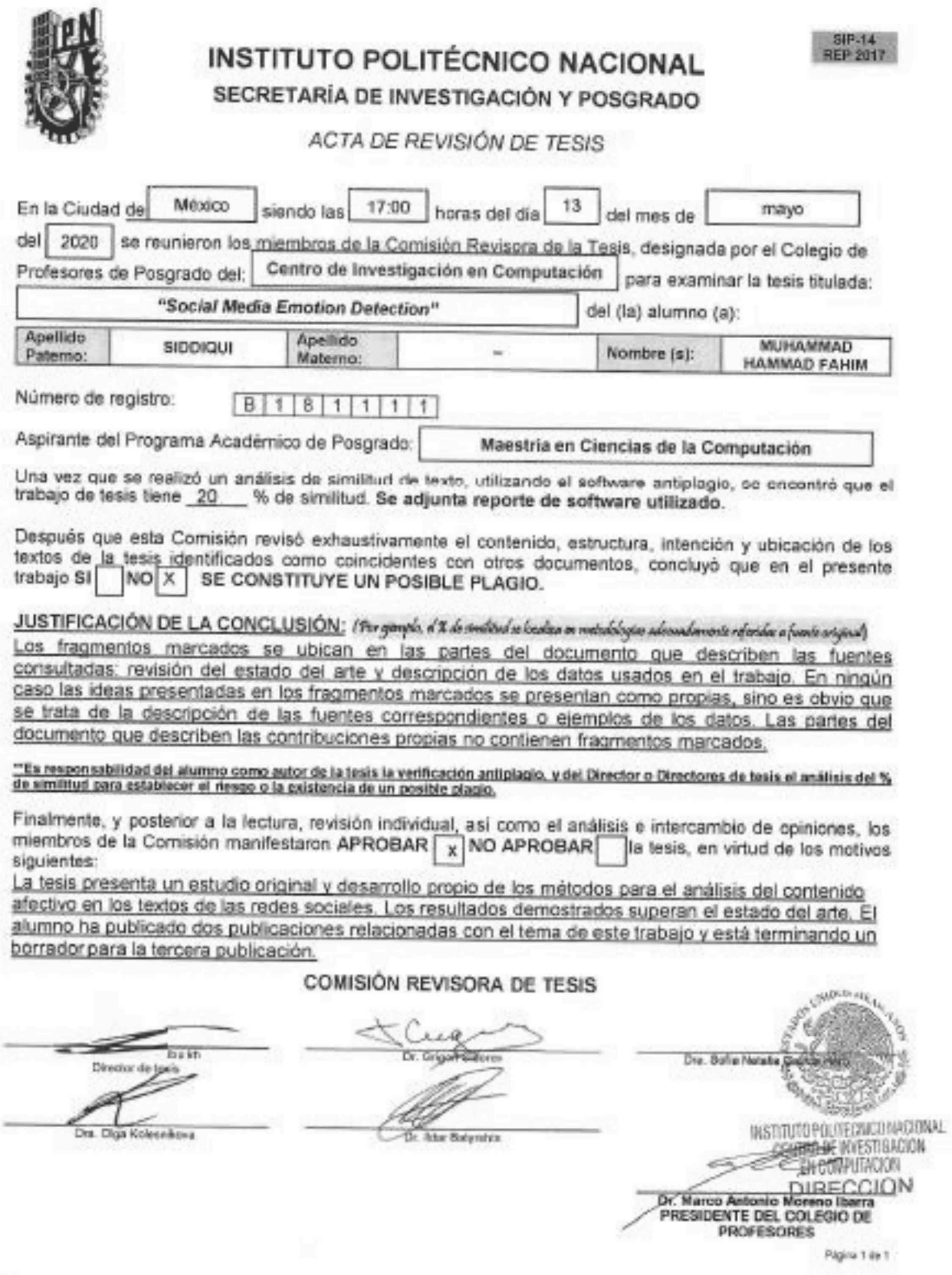
Mr. MUHAMMAD HAMMAD FAHIM SIDDIQUI

Directores de Tesis:

Dr. Alexander Gelbukh

Centro de Investigación
en Computación
Instituto Politécnico Nacional

Junio 2020



SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 17:00 horas del día 13 del mes de mayo del 2020 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado del: Centro de Investigación en Computación para examinar la tesis titulada:

"Social Media Emotion Detection" del (la) alumno (a):

Apellido Paterno:	SIDDIQUI	Apellido Materno:	–	Nombre (s):	MUHAMMAD HAMMAD FAHIM

Número de registro: B 1 8 1 1 1 1

Aspirante del Programa Académico de Posgrado: Maestría en Ciencias de la Computación

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 20 % de similitud. **Se adjunta reporte de software utilizado.**

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** □ **NO** ☒ **SE CONSTITUYE UN POSIBLE PLAGIO.**

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*
Los fragmentos marcados se ubican en las partes del documento que describen las fuentes consultadas: revisión del estado del arte y descripción de los datos usados en el trabajo. En ningún caso las ideas presentadas en los fragmentos marcados se presentan como propias, sino es obvio que se trata de la descripción de las fuentes correspondientes o ejemplos de los datos. Las partes del documento que describen las contribuciones propias no contienen fragmentos marcados.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente, y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** ☒ **NO APROBAR** □ la tesis, en virtud de los motivos siguientes:
La tesis presenta un estudio original y desarrollo propio de los métodos para el análisis del contenido afectivo en los textos de las redes sociales. Los resultados demostrados superan el estado del arte. El alumno ha publicado dos publicaciones relacionadas con el tema de este trabajo y está terminando un borrador para la tercera publicación.

COMISIÓN REVISORA DE TESIS

Dr. Alexander Gelbukh
Director de tesis

Dr. Grigori Sidorov

Dra. Sofía Natalia Galicia Haro

Dra. Olga Kolesnikova

Dr. Ildar Batyrshin

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN

Dr. Marco Antonio Moreno Ibarra
PRESIDENTE DEL COLEGIO DE PROFESORES

Página 1 de 1

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de Mexico el día 01 del mes Junio del año 2020, el que suscribe Muhammad Hammad Fahim Siddiqui alumno del Programa de Maestría en Ciencias de la Computación con número de registro B181111, adscrito al Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Alexander Gelbukh y cede los derechos del trabajo intitulado Social Media Emotion Detection, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección hammad.fahim57@gmail.com, si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Muhammad Hammad Fahim Siddiqui

RESUMEN

Una tarea de clasificación de emociones en PNL se define como la detección y clasificación de las emociones humanas (miedo, tristeza, alegría, sorpresa, etc.) en un texto escrito. El mundo en el que vivimos en este momento les ha brindado a todos la oportunidad de expresar sus sentimientos en varios medios, llamados plataformas sociales. Estas plataformas sociales tienen cientos de millones de usuarios que interactúan entre sí e intercambian contenido (texto, audio, video, etc.). Este contenido contiene emociones expresadas por los humanos, que se desencadenan por varias razones. El objetivo de esta investigación es la detección y clasificación de estas emociones en 11 categorías predefinidas. La clasificación de las emociones es una tarea popular de procesamiento del lenguaje natural (PNL), y se ha trabajado mucho al respecto, pero la tarea que se investiga y presenta en este documento tiene una naturaleza diferente. La investigación realizada ya es principalmente una clasificación binaria, es decir, la detección y clasificación de una emoción en el texto dado. Hay algunos documentos en los que el trabajo se realiza en torno a un total de seis emociones. pero, en esta tarea, lo llevamos más allá a un total de 11 emociones, lo que lo convierte en un problema de clasificación de etiquetas múltiples relativamente difícil.

Las aplicaciones de una tarea como esta van desde el análisis de las emociones del cliente hasta la evaluación de los sentimientos de reacción de las personas después de una noticia política. El conjunto de datos disponible para la tarea fue proporcionado por SemEval-2018 (Mohammad, 2018) y consistió en tweets. Para abordar este problema, hemos realizado una amplia gama de experimentos utilizando el aprendizaje profundo, las atenciones y los enfoques de transferencia de aprendizaje. Se observó un aumento considerable en la métrica de precisión con una nueva técnica de preprocesamiento de datos que incluía características estilísticas en el texto. Dos de los enfoques utilizados en esta investigación fueron capaces de superar la precisión actual (57.4%) en la tarea. Estos dos enfoques incluían un modelo Bi-LSTM con un marco de atención múltiple; Este enfoque fue capaz de lograr una precisión de 58.1% (0.5% de ganancia sobre el estado de la técnica). El otro enfoque utilizaba el aprendizaje por transferencia, entrenando un modelo de Roberta con nuestros datos; Este enfoque fue capaz

de lograr una precisión del 61,2% (ganancia del 3,8% sobre el estado de la técnica). La investigación produjo un nuevo punto de referencia de última generación para la tarea.

ABSTRACT

An emotion classification task in NLP is defined as detection and classification of human emotions (fear, sad, joy, surprise, etc.) in written text. The world that we are living in right now, has given everyone the opportunity to express their feelings in a number of mediums, called social platforms. These social platforms have hundreds of millions of users interacting with each other and exchanging content (text, audio, video, etc.). This content contains emotions in them expressed by humans, which are triggered by a number of reasons. The focus of this research is the detection and classification of these emotions in 11 predefined categories. Emotion classification is a popular Natural Language Processing (NLP) task, and a lot of work has been done around it, but the task that is researched on and presented in this document has a different nature. The research done already is majorly binary-classification, i.e. the detection and classification of one emotion in the given piece of text. There are some papers in which the work is done around a total of six emotions. but, in this task, we take it further to a total of 11 emotions, which makes it a comparatively difficult multi-label classification problem.

The applications of a task like this are wide ranging from customer emotion analysis to gauging the reaction sentiments of people after a political news. The dataset available for the task was provided by SemEval-2018 (Mohammad, 2018) and it consisted of tweets. To approach this problem, we have conducted a wide range of experiments using deep learning, attentions and transfer learning approaches. A considerable increase in the accuracy metric was observed with a new data pre-processing technique that included stylistic features in text. Two of the approaches used in this research were able to surpass the current state-of-the-art accuracy (57.4%) on the task. These two approaches included a Bi-LSTM model with a multiple attention framework; this approach was able to achieve an accuracy of 58.1% (0.5% gain over the state-of-the-art). The other approach used transfer learning, by training a Roberta model with our data; this approach was able to achieve an accuracy of 61.2% (3.8% gain over the state-of-the-art). The research produced a new state-of-the-art benchmark for the task.

Acknowledgment

I would like to pay special thanks to my supervisor **Dr. Alexander Gelbukh** for his continuous technical and intellectual support, guidance and most important his precious time. His cooperation leads me to this success. I would like to appreciate **Dr. Alexander Gelbukh, Dr. Grigori Sidorov**, **Dr. Ildar Batyrshin** and **Dr. Olga Kolesnikova** for serving on my committee. Last but not the least, I would like to thank Consejo Nacional de Ciencia y Tecnología (CONACYT) for providing me scholarship for the duration of my degree which helped me a lot to focus on my studies and achieve this feat.

I would like to admit that I owe all my achievements to my parents, siblings and loved ones who mean the most to me, and whose well wishes have always been a source of determination for me.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Emotion classification

Emotion classification is the identification process of the presence of certain emotions in a given piece of text. Due to the current boom in data creation and the usage of smartphones by people which results in them expressing their thoughts over public social media platforms, such as Twitter, many challenges are being faced by companies and in light of these challenges, and, many opportunities have risen subsequently. The opportunities include the mining of user emotions in written text by companies to get a deeper feedback about their products, or, the general perception of the public about their products. Emotion classification can be of various types, but two major types are listed below:

- **Single label emotion classification** tasks aim to detect and classify a written text towards a certain emotion. This classification identifies if the emotion under question is present in the written text, or not.

- **Multilabel emotion classification** tasks aim to detect and classify the presence of a wide range of emotions in a piece of written text. In the research conducted for this thesis, there was a wide array of emotions (11), to be detected and classified by the models created. There is no compulsion of the presence of all emotions in the text simultaneously, which makes this task more challenging than the simple one. Being a challenging task, it also bears more fruit than its single-label counterpart, which may include the presence of some emotions simultaneously, and their correlation.

Importance and applications of emotion classification

During the past decade, millions of users from around the world have joined social media platforms like Twitter, Facebook, blogging websites and several other online platforms where people come, share and exchange views. These platforms either have a topic of discussion or general discussion and information sharing purpose. The opportunity lies in the volume of data generated by millions of users every day. This data includes their general perspectives, views, statuses or simple, humor. the common property of this shared content is the presence of emotions in it. These emotions represent the current feeling of the user about a specific topic. Now, having a system in place which is effectively detect the presence of emotions in written text, and has the ability to classify it in various categories, can open up a number of applications for organizations. Some applications may include, the emotion mining of people by a company, after the launch of a specific product. Similarly, the system can be used by government organizations to get feedback about the emotional state of people, after passing a law or the results of an election. These being general examples, there can be several other applications where a system like this can be applied and valuable information about user behavior can be extracted.

Motivation

Due the rapid expansion in social media platforms and the exponential growth in the number of users of use these platforms every day, the data generated by these users provides great opportunities for NLP tasks and provides naturally produced and at often times, annotated data for facilitating the tasks. In light of the rich data and opportunities, this research project was conceived in which the emotions expressed by social media users are identified and then classified into 11 emotion types.

The main motivation behind this research was to gain an understanding of the emotions expressed by the users, how they express it and how can we create a model that is automated to classify such data.

Thesis focus

The main aim of this thesis research is to develop new models that can be used to detect and classify emotions in written text. The research starts with exploring the features of emotions in text and then converting those features as machine readable segments so that the computer is able to understand and focus on those segments for successful classification of the correct emotions in the text. For the said purpose text preprocessing is to be used to detect stylistic features in the text and normalize those stylistic features. These stylistic features were then converted to special text embeddings which represent the emphasis of certain text modifications towards the cause of the emotion.

After the text preprocessing, state-of-the-art neural network architectures were used to learn from the text and be able to detect emotions and classify them in unseen texts. This experimental step involved the use of Bi-LSTM neural network architecture on top of which, a fully connected attention layer was deployed to research the improvements and learning mechanism of these deep neural network on the dataset. A number of experiments are done on the neural networks to gauge the behavior on different parameter settings. When a behavior was noted based on parameter tuning, carefully controlled parameters were passed to fine-tune the model in order to get best results possible. The next step was to use transfer learning models for an even higher accuracy benchmark.

A number of pre-trained transfer learning models were fine-tuned on the available dataset. This fine-tuning made the model more aware of the quality of data and the contextual features in the data with respect to the whole input document that contributed towards the triggering of a particular emotion in text. The transfer learning models, by virtue, take the whole document as context to the emotion and learn on the distinctive patterns that cause those emotions.

These models are extremely time consuming to train but, due to their contextual nature, capture more information about the target variable and have the ability to achieve greater accuracies over the dataset.

Problem statement

The aim of this study is to find the stylistic features in informally written text that contribute to the triggering of emotions. These features are then required to be converted to machine readable embeddings that represent that unique style of the written word as an emphasis clause. These embeddings, or, more simply stated: word and sentence representations are to be used in deep-learning and transfer-learning models. These intelligent models are to be fine-tuned on the information being processed by the use of certain fine-tuning parameters. The goal is to obtain the perfect set of parameters which contribute to the best fit of the mode. This goal contributes to attain the final objective, of classifying the input document into correct set of emotion classes.

Scope of study

Emotion classification is a widely studies and researched problem among the researchers of NLP and Computer science. The reason for this wide research is the wide applications of emotion classification in the field of marketing, advertising, government campaigns and general audience feedback. We will extract the features from the written text which contribute to the triggering of a particular emotion. These features will be read and trained on by deep learning and transfer learning algorithms for successful detection and classification of emotions in unseen social media text. Extensive text preprocessing and cleaning will be required and results will be measured by applying the learning models without and without each technique.

General objective

The general objective of this thesis is: To develop deep learning and transfer learning models which classify a social media text (tweet) into a range of 11 emotions, with help of feature extraction using the preprocessing techniques.

Specific objectives

The task aims at finding and learning of the stylistic features in written text, by users on social media platforms. These stylistic features will then be associated with the respective emotions that they cause in the text. The specific objectives, or the timeline of research and development of this thesis is shown below:

1. Explore the problem of emotion classification and the related work that has been done in this field.

2. Collection of data from the data source and convert it to a unique format to be read in and processed by the system.

3. Preprocessing of the data to remove noise from it. Noise may include recurring words, non-words, stop word removal, HTML tags removal, removal of URL links and other writing errors.

4. Preprocessing step to convert stylistic features in the text in form of indicator tags and normalizing the word. This will essentially convert the sentence into a feature tagged document. For example:

   I am veryyyyyy angry on this new government

   will be converted to:

   I am <elongated>very</elongated> angry on this new government.

In the above example, the emphasis on the word very is converted to a feature. Similarly, it is done for all other input documents and for a number of other features, explained in detail later in this document.

5. Converting the extracted features to word embeddings which represent all the words and their associated features (if any).

6. Develop a deep learning model with preset paraments. This will be a Bi-LSTM model for basic analysis

7. Develop a deep learning model with a novel multiple attention architecture, where each attention will pay focus on one single emption type. This architecture will be more complex than the simple one.

8. Train the data on a transfer learning model, which will learn from the contextual features of the document, considering the whole document at once and making sense from the constructs that correspond to the triggers of an emotion. This model is expected to perform much better than the others.

Expected contributions

The research aims to produce the following scientific contributions.

1. Dataset processing for efficient feature extraction and feature annotation on emotion triggering stylistic features.

2. Development of a novel deep learning model with a multiple attention framework for targeted emotion classification.

3. Training of the dataset on a transfer learning model to achieve superior accuracy than the state-of-the-art.

4. Trained models exported for further use and research.

5. Data preprocessing module exported for public use and research.

6. A research publication to be published in a public journal/conference for the researchers to benefit from and work further on.

Thesis outline

Rest of the thesis is organized as follows:

Chapter 2 provides a detailed overview of the research and development work already done in the field of multilabel emotion classification for this dataset. The chapter details the top 9 papers (including the state-of-the-art), that have been published in reputed conferences. In the chapter, a brief discussion about the paper will be done, along with the methodology used by the researchers to approach the problem, and the results they were able to achieve with their implementation. The research work, that achieved state-of-the-art performance is described in much more detail for the reader to analyze the difference in their approach and the approach discussed in this thesis.

Chapter 3 presents a detailed overview of the problem and the dataset to be used in the research.

Chapter 4 presents the proposed approach for multilabel emotion classification. Also explains the preprocessing step in detail and how it effects on the performance of the learning algorithm. This chapter will also include detailed descriptions about the models and architectures used along with their respective hyperparameter tuning and how it affected the final observed results.

Chapter 5 will explain the results achieved by our proposed approached and algorithms, and how these results compare to the current state-of-the-art. There is a discussion section too which details the reasoning of the better results achieved by our implementation and how it differs from the already implemented ones.

Chapter 6 will conclude the thesis, final contributions and explain the possible future work.

Chapter 2

Literature Review on Task 1 of SemEval 2018

Task 1 description

The task <cite1> presented five subtasks in English, Arabic, and Spanish where participants were expected to build systems which automatically determine the intensity of emotions (E) and intensity of sentiment (aka valence V) from the given tweets. It also includes a multi-label emotion classification task for tweets.

EI-reg (Emotion intensity regression subtask):
The task requires you to determine the intensity of emotion E given a tweet that best represents the mental state of the person who tweets. The scale of this task was a real-valued score between 0 (least E) and 1 (most E). Datasets were separately provided for sadness, fear, anger and joy.

EI-oc (Emotion intensity ordinal classification subtask):
The task requires you to classify the tweet into one of four ordinal classes of intensity of E that best represents the mental state of the person who tweets, given a tweet and an emotion E. Datasets were separately provided for sadness, fear, anger and joy.

V-reg (Sentiment intensity regression subtask):
The task requires you to determine the intensity of sentiment or valence (V) that best represents the mental state of the person who tweets, given a tweet, with a real-valued score between 0 (most negative) and 1 (most positive).

V-oc (Sentiment analysis, ordinal classification, subtask):

The task requires you to classify the tweet it into one of seven classes, corresponding to the positive and negative sentiment intensity, with the best representation of the mental state of the tweeter.

E-c (Emotion classification subtask):
The task requires you to classify it as 'neutral or no emotion' or as one or multiple of the eleven given emotions that best represent the mental state of the person who tweets. Here, E refers to emotion, OC refers to ordinal classification, C refers to classification, V refers to valence or sentiment intensity, REG refers to regression, EI refers to emotion intensity.

ELiRF-UPV at SemEval-2018 Tasks 1 and 3: Affect and Irony Detection in Tweets

In this paper (González & Pla, 2018) the author tackles two tasks where tasks one required them to automatically classify the intensity of emotions and sentiment or valence from the given tweets. The second task required them to a) perform binary classification task to predict whether a tweet is ironic or not b) perform multi-class classification task to predict four labels (verbal irony realized through a polarity contrast, verbal irony without such a polarity contrast, descriptions of situational irony and non-irony). The combined deep learning-based system that assembles CNN and LSTM neural networks for both the task with some slight changes. They achieved the accuracy of 0.552, Micro average F1 of 0.658 and Macro average F1 of 0.512 on multi-label emotion classification(E-c).

YNU-HPCC at SemEval-2018 Task 1: BiLSTM with Attention Based Sentiment Analysis for Affect in Tweets

This paper (Zhang & Zhang, 2018) was submitted as a result of the SemEval-2018 competition where they tackled Task 1. Task 1 named Affect in Tweets had five subtasks in English and Spanish. The authors tackled all subtasks involving prediction of emotion or sentiment intensity and determining multi label emotion classification. BiLSTM was used in order to extract bi-directional word information. Contribution of each word was determined by the attention mechanism for improving the scores. BiLSTM with an attention mechanism was also aided by a few deep-learning algorithms with the suitability of each subtasks. Regression and ordinal classification tasks, we dealt with the use of domain adaptation and ensemble learning methods to leverage the base model, whereas a multi-label task was dealt with single base model. The results achieved for the multi label emotion classification(E-c) using the approach was the accuracy of 0.558, micro average F1 of 0.674 and macro-average F1 of 0.488.

TeamUNCC at SemEval-2018 Task 1: Emotion Detection in English and Arabic Tweets using Deep Learning

This paper (Abdullah & Shaikh, 2018) countered all the subtasks with only one architecture which is a novelty in both English and Arabic. The input was given as a combination of word2vec and doc2vec embeddings and a set of psycholinguistic features extracted from Affective Tweets Weka-package. They modeled a fully connected neural network architecture. The network architecture consists of Dense network and LSTM- network. The built LSTM network consists of 256 neurons that connects to two hidden layers and two dropouts (0.3, 0.5). For optimization, they used SGD optimizer. The first hidden layer has 256 neurons, while the second layer has 80 neurons. Both layers use the RELU activation function. The output layer consists of one sigmoid neuron, predicting the intensity of the emotion or the sentiment between 0 and 1. Whereas, the Dense Network has the input of 445-dimensional vector feeding into a fully connected neural network with three dense hidden layers. The activation function for each layer is RELU. The output layer consists of one sigmoid neuron, predicting the intensity of the emotion between 0 and 1. The results they obtained for the Multi-Label Emotion Classification task were the accuracy of 0.471.

Amobee at SemEval-2018 Task 1: GRU Neural Network with a CNN Attention Mechanism for Sentiment Classification

The paper (Rozental & Fleischer, 2018) tackles the Multi-label emotion classification subtask of Task-1 in a novel way. They used Long Short-term Memory (LSTM) networks, including LSTM with attention mechanism and bidirectional LSTM (BiLSTM). They performed transfer learning by first pre-training the LSTM networks on sentiment data and then concatenated the penultimate layers into a single vector feeding new dense layers. For the Multi-label emotion classification subtask, they utilized hierarchical clustering to group correlated emotions collectively and then trained the same model incrementally for emotions within the same cluster unit. The novel method claims to outperform the system which trains on each emotion independently. They achieved the accuracy of 0.566 in the multi-label emotion classification task.

TCS Research at SemEval-2018 Task 1: Learning Robust Representations using Multi-Attention Architecture

In this paper (Meisheri & Dey, 2018) while tackling Task 1, they combined three different features generated using deep learning techniques and support vector machine to develop a unified ensemble system. The tweet is trained using a multi-attention-based architecture which take the input of different pre-trained embeddings. For the multi-label emotion classification task, the output layer of the used deep learning model contained eleven neurons and sigmoid as an activation function. Binary cross entropy was used as a loss function with Stochastic gradient descent with Nesterov momentum, 10−6 learning rate decay as optimizer and 0.01 learning rate. The presented ensemble system was capable of handling noisy sentiment dataset over multi-label dataset. The mixture of embedding in parallel made the system unique as it generated better representations with respect to sentiment. They were able to achieve second rank in the multi-label emotion classification task with the accuracy of 0.582, the micro- average F1 of 0.694 and the macro- average F1 of 0.534.

FOI DSS at SemEval-2018 Task 1: Combining LSTM States, Embeddings, and Lexical Features for Affect Analysis

The paper (Karasalo & Bolin, 2018) mentions the results and methodology of the English language datasets in Task 1 only. They used transfer learning from LSTM nets trained on large sentiment datasets combined with lexical features and embeddings. The paper mentions three different methods for feature extraction; one utilizing the Weka Affective Tweets package and two using variants of Long Short-Term Memory (LSTM) nets obtained by training on large sentiment datasets. The activation function for the two hidden layers was tanh and for the output layer were set to sigmoid for the E-c subtask. For the E-c subtask Adam optimizer was used for the classification with the binary cross-entropy loss. They used L2-regularization on the parameters of the hidden layers. For each subtask, the hyperparameters of the neural network were found by a grid search evaluating the PCC on the validation data. The results obtained for the multi-label emotion classification were 0.554 in the accuracy, 0.674 micro average F1 and 0.490 of macro average F1.

Tw-StAR at SemEval-2018 Task 1: Preprocessing Impact on Multi-label Emotion Classification

The paper (Mulki & Babaoglu, 2018) solely tackles the subtask Multi-label emotion classification (E-c) of SemEval 2018 Task 1 in Arabic, English and Spanish tweets. The used the binary relevance transformation strategy and the tweets features ere generated using TF-IDF scheme. They examined several single and combinations of pre-processing tasks to enhance the performance which was proven with the results. For the Arabic tweet's dataset, ISRI stemmer was used which improved the accuracy by 5.1% percentage points. This change was considered to work as the ISRI can now handle wider range of Arabic vocabulary as it gets normalized form of words having no stem. On the contrary stemming behaved differently when it was applied on both English and Spanish tweets as it slightly increased the accuracy by 0.3% and 0.8%. They combined emoji tagging with lemmatization and stop words removal to achieve the best performances with a micro average F-measure of 60.6% and 52.3% for English and Spanish respectively. The best results obtained by the system were the 0.499(Accuracy), 0.58(Micro F1 average) and 0.444(Macro F1) in Arabic, 0.48(Accuracy), 0.606(Micro F1 average) and 0.46(Macro F1) in English and lastly 0.431(Accuracy), 0.523(Micro F1 average) and 0.413(Macro F1) in Spanish.

NEUROSENT-PDI at SemEval-2018 Task 1: Leveraging a Multi-Domain Sentiment Model for Inferring Polarity in Micro-blog Text

The paper (Dragoni, NEUROSENT-PDI at SemEval-2018 Task 1: Leveraging a Multi-Domain Sentiment Model for Inferring Polarity in Micro-blog Text, 2018) tackles the Task 1 in only English tweets with the same approach but slight differences based on each subtask. They used a supervised approach that builds on word embeddings and neural networks. Word embeddings were built and then the tweets were converted in the corresponding vector representation and given as input to the neural network. The goal of this was of learning the different semantics contained in every emotion that best represent the mental state of the tweet's author. The system was required to classify the tweet as neutral, no emotion or as one, or multiple, of eleven given emotions. The output layer of our neural network is built by eleven neurons implementing the SIGMOID activation function. The results obtained for the multi-label emotion classification were 0.192 in the accuracy, 0.318 micro average F1 and 0.268 of macro average F1.

State of the Art

NTUA-SLP at SemEval-2018 Task 1: Predicting Affective Content in Tweets with Deep Attentive RNNs and Transfer Learning

This paper (Dragoni, NEUROSENT-PDI at SemEval-2018 Task 1: Leveraging a Multi-Domain Sentiment Model for Inferring Polarity in Micro-blog Text, 2018) worked only on English tweets dataset of Task 1. The author proposed a Bi-LSTM architecture equipped with a multi-layer self-attention mechanism. They were able to get the best results in the subtask of Multi-label Emotion classification. They used word2vec word embeddings trained on a large collection of 550 million Twitter messages, augmented by a set of word affective features. They used the word2vec algorithm, with the skip-gram model, negative sampling of 5 and minimum word count of 20. Word2vec word embeddings added 10 affective dimensions to initialize the first layer of our neural networks. To build the pretrained model, they initialized the weights of the embedding layer with the embeddings and trained a bidirectional LSTM (BiLSTM) with a deep self-attention mechanism. Afterwards, they utilized the encoding part of the network (BiLSTM and the attention layer) throwing away the last layer. This pretrained model was used for all subtasks, with the slight changes to subtask specific final layer for classification/regression. The attention mechanism improved the model performance and allowed them gain insights into the models and to identify salient words in tweets.

They experimented with two fine-tuning techniques. The first approach was to fine-tune the whole network including both the pretrained encoder (BiLSTM) and the task-specific layer. The second method was to use the pre-trained model only for weight initialization, freeze its weights during training and just fine-tune the final layer. The first approach showed promising results in all tasks.

For model regularization they added Gaussian noise to the embedding layer, making the model more robust to overfitting. In addition to that, we use dropout and we stop training after the validation loss has stopped decreasing.

They used Adam algorithm for optimizing our networks, with minibatches of size 32 and they clipped the norm of the gradients at 1, as an extra safety measure against exploding gradients. They also applied class weights to the loss function, penalizing more the mis-classification of under-represented classes. These weights were computed as the inverse frequencies of the classes in the training set. In order to tune the hyperparameter of their model, they adopted a Bayesian optimization approach, performing a more time-efficient search in the high dimensional space of all the possible values, compared to grid or random search. They set size of the embedding layer to 310 (300 word2vec + 10 affective dimensions), which they regularized by adding Gaussian noise with σ = 0.2 and dropout of 0.1. The sentence encoder was composed of 2 BiLSTM layers, each of size 250 (per direction) with a 2- layer self-attention mechanism. Lastly, they applied dropout of 0.3 to the encoded representation. Jaccard index is used for the multi-label classification subtask (E-c).

In the Emotion multi-label classification subtask (E-c), transfer learning was unable to outperform the random initialization model. This can be because the source dataset was not diverse enough to boost the model performance when classifying the tweets into none, one or more of a set of 11 emotions. As for fine-tuning or freezing the pretrained layers, the overall results demonstrated that enabling the model to fine-tune always resulted in significant gains. This reassured allowing the weights of the model to adapt to the target dataset, hence, encoding task-specific information, results in performance gains.

The results achieved in the Multi-label Emotion Classification were the accuracy of 0.595, micro average F1 of 0.709 and macro average F1 of 0.542.

Chapter 3

Multi-label Emotion Classification Problem

Emotions are central to language and are the key to people's feelings and thoughts. Humans are known to perceive hundreds of different emotions. One can feel multiple emotions at one point and can also be completely neutral with the emotions. Multi-label classification originated from the text categorization problem, where each document simultaneously belonged to several predefined topics. In multi-class classification the classes are mutually exclusive, however in a multi-label problem each label represents a different classification task, but the tasks are related to each other. For example, multi-class classification makes the assumption that each sample is assigned to one and only one label: a vegetable can be either cabbage or a brinjal but not both at the same time. Whereas, an instance of multi-label classification can be that a text might be about any of emotion simultaneously or none of these.

Multi-label emotion classification problem has attracted considerable interest in the research community due to its applicability to a wide range of domains, including text classification. Now as that is established now, Multi-Label Emotion Classification that was tackled in the thesis aimed to develop an automatic system to determine the existence of the emotion in a text out of eleven emotions: the eight Plutchik categories (joy, sadness, anger, fear, trust, disgust, surprise, and anticipation) with the inclusion of common emotions in tweets(love, optimism, and pessimism). Given a tweet, the task was to classify it as 'neutral or no emotion' or as one, or more if needed, of eleven given emotions that best represent the mental state of the person who is tweeting.

Evaluation Measures

The standard evaluation measure used by the SemEval task for multi-label emotion classification was Accuracy, Micro average F1 and Macro average F1 and is also the common practice for the defined task for evaluation. Hence, it is very important to understand what these evaluation metrics mean and why are they so essential for the task. The evaluation measures for single-label classification are usually different than for multi-label classification. Here in single-label classification we use simple metrics such as precision, recall, accuracy, etc., However, in a multi-label classification such a s emotion classification especially, a misclassification is no longer a hard wrong or right. In this case prediction containing a subset of the actual classes should be considered more appropriate than a prediction that contains none of them, i.e., predicting two of the three emotion labels correctly is better than predicting no emotion label at all. To measure a multi-label classifier, we have to average out the classes in some way. The two different methods of doing this are called micro-averaging and macro-averaging. It is important to note that Macro-averaging and Micro averaging assume equal weights for labels and examples respectively. Hence, it is not difficult to show that both the equation bel0w holds.

Accuracy macro(h) = Accuracy micro(h) and Accuracy micro(h) +hloss(h) = 1

Accuracy

For the accuracy each tweet can have one or more gold emotion labels, and one or more predicted emotion labels. Multi-label accuracy in this case will be defined as the size of the intersection of the predicted and gold label sets divided by the size of their union. This measure is calculated for each tweet *t*, and then is averaged over all tweets in the dataset *T as shown in the diagram below*:

$$Accuracy = \frac{1}{|T|}\sum_{t \in T}\frac{|G_t \cap P_t|}{|G_t \cup P_t|}$$

Gt: is the set of the gold labels for tweet *t*

Pt: is the set of the predicted labels for tweet *t*

T: is the set of tweets

Hence, the metrics here will be example based.

Micro Average F1

In micro-averaging all True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for each class are summed up and then the average is taken. The equation is mentioned below. Micro-averaged F-score is calculated as follows:

$$Micro\text{-}avg\ Precision\ (micro\text{-}P) = \frac{\sum_{e \in E} number\ of\ tweets\ correctly\ assigned\ to\ emotion\ class\ e}{\sum_{e \in E} number\ of\ tweets\ assigned\ to\ emotion\ class\ e}$$

Figure 1 Micro average Precision

$$Micro\text{-}avg\ Recall\ (micro\text{-}R) = \frac{\sum_{e \in E} number\ of\ tweets\ correctly\ assigned\ to\ emotion\ class\ e}{\sum_{e \in E} number\ of\ tweets\ in\ emotion\ class\ e}$$

Figure 2 Micro average Recall

$$Micro\text{-}avg\ F = \frac{2 \times micro\text{-}P \times micro\text{-}R}{micro\text{-}P + micro\text{-}R}$$

Figure 3 Micro Average F1

General Example:

$$\text{Microaveraging Precision } Prc^{micro}(D) = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FPs(c_i)}$$

$$\text{Microaveraging Recall } Rcl^{micro}(D) = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FNs(c_i)}$$

Figure 4 Micro Averaging Precision and Recall general equation

For the micro-averaging method, you sum up the individual true positives, false positives, and false negatives of the system to apply Precision and Recall functions as mentioned in the formula given above. And then the micro-average F1-Score will be simply be the harmonic mean of above two equations.

Macro Average F1

For the Macro Average we simply take the average of the precision and recall of the system on different emotion sets. A macro-average will compute the metric independently for each class treating all classes equally. Macro-averaged F-score is calculated as follows:

$$Precision\ (P_e) = \frac{number\ of\ tweets\ correctly\ assigned\ to\ emotion\ class\ e}{number\ of\ tweets\ assigned\ to\ emotion\ class\ e}$$

Figure 5 Macro Average Precision

$$Recall\ (R_e) = \frac{number\ of\ tweets\ correctly\ assigned\ to\ emotion\ class\ e}{number\ of\ tweets\ in\ emotion\ class\ e}$$

Figure 6 Macro Average Recall

$$Macro\text{-}avg\ F = \frac{1}{|E|} \sum_{e \in E} F_e$$

Figure 7 Macro Average F1

General Example:

- Macroaveraging Precision $Prc^{macro}(D) = \frac{\sum_{c_i \in \mathcal{C}} Prc(D, c_i)}{|\mathcal{C}|}$

- Macroaveraging Recall $Rec^{macro}(D) = \frac{\sum_{c_i \in \mathcal{C}} Rcl(D, c_i)}{|\mathcal{C}|}$

Figure 8 Macro Average precision and recall general example

Macro-averaging is especially useful when you want to know how the system performs overall across the sets of data. One should not come up with any specific conclusion with this average. On the contrary, micro-averaging can be a useful measure when your dataset varies in size.

Hamming-Loss

Hamming-Loss can be defined as the fraction of labels that are incorrectly predicted, i.e., the fraction of the wrong labels to the total number of labels. The general equation can be seen below:

$$\frac{1}{|N| \cdot |L|} \sum_{i=1}^{|N|} \sum_{j=1}^{|L|} \operatorname{xor}(y_{i,j}, z_{i,j}), \text{ where } y_{i,j} \text{ is the target and } z_{i,j} \text{ is the prediction.}$$

Figure 9 Hamming Loss equation

Dataset: SemEval Task 1 (Affect in Tweets)

Multi-Label Emotion Tweets Dataset is collected to verify the presence/absence of 11 emotions. The eleven emotions include anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust and lastly neutral or no emotion.

Annotating Tweets

The dataset was annotated by crowdsourcing. Around 5% of the tweets were annotated internally beforehand by the authors which were referred to as gold tweets. The gold tweets were interspersed with other tweets and if a crowd-worker got a gold tweet question wrong, they were immediately notified of the error. If the annotators accuracy on the gold tweet questions fell below 70%, they were refused further annotation, and all of their annotations were discarded. This ensured the quality of the annotation and served as a mechanism to avoid malicious annotations.

Tweet were presented one at a time to the annotators and they were asked that which of the following options best described the emotional state of the tweeter out of the following options:

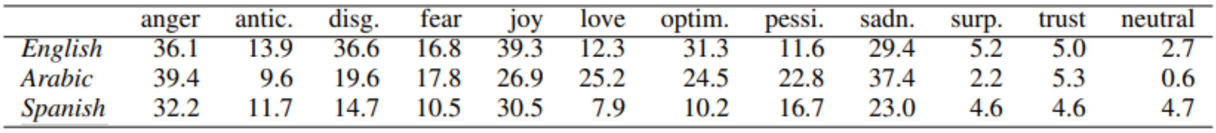
- anger (also includes annoyance, rage)
- anticipation (also includes interest, vigilance)
- disgust (also includes disinterest, dislike, loathing)
- fear (also includes apprehension, anxiety, terror)
- joy (also includes serenity, ecstasy)
- love (also includes affection)
- optimism (also includes hopefulness, confidence)
- pessimism (also includes cynicism, no confidence)
- sadness (also includes pensiveness, grief)

- surprise (also includes distraction, amazement)
- trust (also includes acceptance, liking, admiration)
- neutral or no emotion

Example tweets were provided to the annotators in advance with examples of suitable responses for guidance and clarification. The gold tweets were set up, they were annotated by more than seven people, however, he median number of annotations was seven. In total, 303 people annotated between 10 and 4,670 tweets each. A total of 174,356 responses were obtained.

Given that 25% of the responses (two out of seven people) indicated that a certain emotion applies, then that label was selected. They refer to this aggregation as Ag2 in the paper. Given that no emotion received at least 40% of the responses (three out of seven people) and more than 50% of the responses indicated that the tweet was neutral, then the tweet was marked as neutral. Vast majority of the cases, had tweet labeled either as neutral or with one or more of the eleven emotion labels.

Training, Development, and Test Sets



	anger	antic.	disg.	fear	joy	love	optim.	pessi.	sadn.	surp.	trust	neutral
English	36.1	13.9	36.6	16.8	39.3	12.3	31.3	11.6	29.4	5.2	5.0	2.7
Arabic	39.4	9.6	19.6	17.8	26.9	25.2	24.5	22.8	37.4	2.2	5.3	0.6
Spanish	32.2	11.7	14.7	10.5	30.5	7.9	10.2	16.7	23.0	4.6	4.6	4.7

Figure 10 Distribution of emotions in the dataset

The dataset contains a total of 10,983 tweets. The tweets were divided into train, dev and test classes with 6,838 tweets in training set, 886 tweets in Dev, 3,259 tweets in test set. Above mentioned image shows the percentage of tweets that were labeled with a given emotion (after aggregation of votes).

Chapter 4

Approach

Chapter description

In this chapter, a detailed explanation of the different approaches used, will be presented. these approaches will contain as much detail possible to fully explain the processes, the architectures and the outcomes. The chapter will start with preprocessing and will then move on to complex models which were used to solve the problem. These models will include the simple BiLSTM architecture, the Bi-LSTM architecture will single attention layer, the Bi-LSTM architecture with a multiple attention layer architecture, and finally the transfer learning approach.

Preprocessing and feature extraction

The first step after gathering and setting up the dataset is to read the data and make it eligible for machine to understand and learn from it. This process is called preprocessing. It varies from problem to problem and in this section, the preprocessing steps used in the research are presented.

Text preprocessor

This is a distinct module in our approach which is responsible to process the noisy text of social media tweets. The preprocessing steps, and their detailed descriptions performed by this module are provided below:

1. **Normalization**

   In this step, the tweet is normalized and many other textual features present in it, apart from normal dictionary words are converted into one form for the machine to understand. There are 9 types of normalizations done on the input text. These are described below:

- *URL*: This normalization step processes a given URL in the text. The URL is just a website address and does not contain any meaning to it. Hence, the URL in the text is replaced by a <URL> tag. This URL tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for URL normalization in our approach is shown below.

	Tweet
Before	I love the new Brad Pitt movie https://www.imdb.com/name/nm0000093/
After	I love the new Brad Pitt movie <URL>

Table 1 URL cleaning

- *Email*: This normalization step processes a given email address in the text. The email is just a mailing address and does not contain any meaning to it. Hence, the email address text is replaced by an <email> tag. This email tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for email normalization in our approach is shown below.

	Tweet
Before	sorry to hear about your experience, however, please do not hesitate to contact us via live chat or email at askdysonUS@dyson.com
After	sorry to hear about your experience , however , please do not hesitate to contact us via live chat or email at <email>

Table 2 Email cleaning

- *Number*: This normalization step processes a given number in the text. The number is just a digit and does not contain any emotional meaning to it. Hence, the numbers replaced by a <number> tag. This number tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for number normalization in our approach is shown below.

	Tweet
Before	you charge 150 extra for sending someone out and your cable service still doesn't work.
After	you charge <number> extra for sending someone out and your cable service still does not work .

Table 3 Number Cleaning

- *Money*: This normalization step processes a given monetary object in the text. The currency is just a sign and does not contain any emotional meaning to it. Hence, the currency objects replaced by a <money> tag. This money tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for currency normalization in our approach is shown below.

	Tweet
Before	So far ours greet have raised £250 for Trump with more to come in.
After	So far ours greet have raised <money> for Trump with more to come in.

Table 4 Currency cleaning

- *Phone*: This normalization step processes a given phone number in the text. The phone number is just a series of digits and does not contain any emotional meaning to it. Hence, the phone number is replaced by a <phone> tag. This phone tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for phone number normalization in our approach is shown below.

	Tweet
Before	Get ready for tonight with personal hydration. Book now. 305-912-4937.
After	get ready for tonight with personal hydration . book now . <phone>

Table 5 Phone number cleaning

- *User*: This normalization step processes a given username in the text. The username is just a name of the account user and does not contain any emotional meaning to it. Hence, the username is replaced by a <user> tag. This user tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for username normalization in our approach is shown below.

	Tweet
Before	@Max_Kellerman it also helps that the majority of NFL coaching is inept.
After	<user> it also helps that the majority of NFL coaching is inept .

Table 6 Username cleaning

- *Time*: This normalization step processes a given time in the text. The time is just a numerical object and does not contain any emotional meaning to it. Hence, the time is replaced by a <time> tag. This time tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for time normalization in our approach is shown below.

	Tweet
Before	Awake at 5.30am with a seriously bad throat 😩😩😩😩😷🤒
After	awake at <time> with a seriously bad throat 😩 😩 😩 😩 😷 🤒

Table 7 Time cleaning

- *Date*: This normalization step processes a given date in the text. The date is just a numerical/string object and does not contain any emotional meaning to it. Hence, the date is replaced by a <date> tag. This date tag will have a distinct embedding to be fed into the learning model. Removing it, will remove the noise from our data. An example for date normalization in our approach is shown below.

	Tweet
Before	tells me my order will ship on Sept 12, arriving by Sept 19.
After	tells me my order will ship on <date> , arriving by <date> .

Table 8 Date cleaning

2. **Annotations for emotions and emotion causing features**

Social media text is very informal. Due to limited scope of writing, people tend to use various writing styles to express their feelings and emotions. General word normalization and data cleaning gets rid of these writing styles and for a task like emotion detection, where much of the meaning is present in the writing styles, information is lost. To counter these problems, preserve stylistic information and still be able to normalize the words to their dictionary representations, this module is developed, and does the following processing steps on the data.

- Hashtag: Hashtags are widely used in social media texts. these tags represent a state, dilemma or just a taunt to someone or some organizations. These hashtags contain important information related to emotional state of the user writing it. We preprocess hashtags by normalizing them to words and annotating a hashtag, tag around them for the machine to make sense from it. Following is an example annotation for hashtags.

	Tweet
Before	“Worry is a down payment on a problem you may never have'. Joyce Meyer. #motivation #leadership #worry
After	worry is a down payment on a problem you may never have ' . joyce meyer . <hashtag> motivation </hashtag> <hashtag> leadership </hashtag> <hashtag> worry </hashtag>

Table 9 Hashtag cleaning and normalization

- All caps: It is also a very common practice to express emphasis on a certain topic by using all capital words. These all capital words often express anger, or joy. There is need to capture this detail in our processing so this information does not get lost. To serve the purpose, a special all caps tag is places in front and rear of the word before normalizing it to its normal un-capped state. An example of all cap’s annotation is shown below.

	Tweet
Before	my cat is bloody lucky the <allcaps> rspca </allcaps> were not open
After	My cat is bloody lucky the RSPCA weren't open

Table 10 Case normalization

- Elongated: Writing and expanded version of a word is also a very common practice in social media information writing. There are often cases where the user tries to explain the importance of something, goes short of adjectives and use an elongated version of the word. This also contains expressive and emotional information which will be lost if the word is normalized back to its dictionary equivalent. In this preprocessing step, an elongated tag was added before and after the word for information persistence. An example of elongated annotation is shown below.

	Tweet
Before	with a seriously bad throat 😩😩😩😩😷🤒 glands feel huuuuuuuge
After	with a seriously bad throat 😩 😩 😩 😩 😷 🤒 glands feel huge <elongated>

Table 11 Elongated word normalization

- Repeated: Writing repeating instances of words or characters is also a very to express strong reactions in social media text. This repetends will get lost after normalization and should be preserved as it is an important metric for emotions. To do this, a repeated tag is placed before and after the word to preserve the information. An example of repeated annotation is shown below.

	Tweet
Before	I will beat you !!! Always thought id be gryffindor so this is a whole new world for me
After	i will beat you ! <repeated> always thought id be gryffindor so this is a whole new world for me

Table 12 Repeated character normalization

- Emphasis: To express emphasis on a certain word, people often try to enclose it in a pair of asterisks. These asterisks show that the user is trying to emphasize on this word. Again, information will be lost if these words are normalized and these asterisks are removes. A special emphasis tag is placed after such words in the text. An example of this annotation is shown below.

	Tweet
Before	They will be a formidable challenge' Roy Hodgson on Northampton Town *shudders*
After	they will be a formidable challenge ' roy hodgson on northampton town shudders <emphasis>

Table 13 Emphasis normalization

- Censored: People tend to express anger on social media platforms with use of abusive language and because of the platform laws, or public audience, they try to censor the abusive word which conveys the meaning and essentially does not show the whole word. In normalization, since it is not a dictionary word, it can be removed. If removed, emotion related information is lost. To process this, the word is normalized to the dictionary word and a censored tag is placed with it. A censored annotation example is shown below.

	Tweet
Before	Instagram seriously sort your sh*t out. I spent ages writing that caption for you to delete it and not post it!
After	instagram seriously sort your sh*t <censored> out . i spent ages writing that caption for you to delete it and not post it !

Table 14 Censored word normalization

- Emotional annotations: Social media text is very rich in terms of emotion expressions. Along with the stylistic features discussed above, many people use special characters to make emotion expressing faces in their text. These faces are widely called as emoticons/emojis. Just removing these faces because they are not words, will lose a lot of important emotional information from the text. To overcome this, special tags for emotions are placed for these faces so the machine can learn their importance in shaping the emotion of the whole sentence. Examples of emotional annotations are shown below.

  - Happy:

	Tweet
Before	You've got quite the cheery young Souls fan. :)
After	you have got quite the cheery young souls fan . <happy>

Table 15 Emoji tagging - happy

- Annoyed

	Tweet
Before	Why is it always me picking up the pieces :/
After	why is it always me picking up the pieces <annoyed>

Table 16 Emoji tagging - annoyed

- Laugh

	Tweet
Before	not going to waste my energy holding a grudge against someone who wasnt even in my life a year XD
After	not going to waste my energy holding a grudge against someone who wasnt even in my life a year <laugh>

Table 17 Emoji tagging - laugh

- Tongue sticking out

	Tweet
Before	True. We were rejoicing the fact that we signed a quality young CB but we need one more at least. @MarcBartra ? mebbe :P
After	true . we were rejoicing the fact that we signed a quality young cb but we need one more at least . <user> ? mebbe <tong>

Table 18 Emoji tagging - tongue sticking out

- Wink

	Tweet
Before	'shit' doesn't even begin to describe these fiery little demons straight from hell ;)
After	' shit ' does not even begin to describe these fiery little demons straight from hell <wink>

Table 19 Emoji tagging - wink

3. Unpacking contractions

It is a rather common occurring in English that people use contractions in their writing to shorten the common words in order to occupy less space and sounding more efficient. This puts forwards a great deal of noise where two users might have used to same words, while one with contractions and the other not with contractions. It is important to unpack these contractions so the data becomes consistent and is easier for the machine to learn from it. For the said purpose, unpacking on contractions was done. The following table illustrates the contractions and their respective unpacking done in the text.

Contraction	Unpacking
ain't	am not / are not
aren't	are not / am not
can't	Cannot
can't've	cannot have
could've	could have
couldn't've	could not have
didn't	did not
doesn't	does not
don't	do not
hadn't	had not
hadn't've	had not have
hasn't	has not
doesn't	does not
don't	do not
hadn't	had not
i'll've	I shall have / I will have
i'll	I shall / I will
i'd've	I would have
i'd	I had / I would

how's	how has / how is
how'll	how will
how'd'y	how do you
how'd	how did
he's	he has / he is
he'll've	he shall have / he will have
he'll	he shall / he will

Table 20 Word contraction unpacking

Word Embeddings

GloVe (Global Vectors for Word Representation) (Pennington, 2014) is a word vector technique putting all the matched words in the vector space, where similar words cluster together in the form of embedding and different words repel. It is a global log-bilinear regression model for the unsupervised learning of word representations that outperforms other models on tasks of word similarity, word analogy, and named entity recognition. The benefit of GloVe is that, unlike Word2vec, it does not rely just on local context information of words (i.e. the semantics learnt for a given word, is only affected by the surrounding words), but incorporates global statistics (word co-occurrence) to obtain word vectors.

Both global and local statistics are very important since each type of statistic has their own advantage. Explaining the embedding, let's take two words i and j that portray a particular aspect of interest; for concreteness, assume we are interested in the concept of thermodynamic phase, for which we might take i = ice and j = steam. The relationship of these words can be examined by the ratio of their co-occurrence probabilities with various probe words, k. For words k is related to ice but not steam, say if k = solid, we would expect the ratio Pik /Pjk to be large. Similarly, for words k related to steam but notice, say k = gas, the ratio should be small. Moreover, for words k like water or fashion, that are either related to both ice and steam, or to none of them, the ratio should be close to one. In GloVe compared to the raw probabilities, the ratio is better able to distinguish relevant words (solid and gas) from irrelevant words (water and fashion) and it is also better able to distinguish between the two relevant words.

The paper of GloVe mainly contributes in three important questions.

- We don't have an equation, e.g. F (i, j, k) = P_ik/P_jk, but just an expression (i.e. P_ik/P_jk).
- Word vectors are high-dimensional vectors, however P_ik/P_jk is a scalar. So, there's a dimensional mismatch.
- There are three entities involved (i, j and k). But computing loss function with three elements can get hairy, and needs to be reduced to two.

The GloVe model performs significantly better than the other baselines, often with smaller vector sizes and smaller corpora. The results using the word2vec tool as the paper mentions are better than most of the previously published results as per the date of the published paper. This is due to a number of factors, including the choice to use negative sampling, the number of negative samples, and the selection of the corpus.

For our experiments we used Glove Embeddings for Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors). Since our emotion dataset was extracted from Twitter, GloVe embeddings for twitter was the perfect fit for the task as it tackles social media informal text decently.

Model 1: BiLSTM

LSTM stands for Long Short-Term Memory. It belongs to the family of recurrent neural networks because of its ability to reiterate over and over again on the test set with special memory blocks to preserve information in them for further use. In recent years, LSTM's have gained popularity amongst the research community, because of the ability of the model to perform much better than machine learning algorithms, or even than their normal deep learning counterparts. For the first experiment in the research, a BiLSTM network was developed and a soft max function was applied on top of it to get the results.

In this supervised problem of emotion classification, we classified the Twitter tweets by using Recurrent Neural Network (RNN), to be precise, we implemented LSTM and Bi-LSTM to model the tweets (Ying Zeng, 2016). LSTM initially introduced by (Schmidhuber, 1997) has demonstrated that LSTM can resolve long-time hard problems, for example, machine translation and speech recognition. Bi-LSTM (Alex Graves, 2013) is a continuation of traditional LSTM to train both past and future information on the input sequence at each time step. The second LSTM is a copy of the first LSTM but reversed so that we can leverage both backward and forward input features. Both networks, traditional LSTM, and Bi-LSTM trained by using back-propagation (Huo, 2016). After the generation of word vectors in the embedding layer, the sequence of vectors provided to a traditional single-LSTM layer or Bi-LSTM layer. The number of neurons is set up as 64 to maintain the coherence of dimensions for both the LSTM Layer and the Bi-LSTM Layer. The Figure 2 presenting a Bi-LSTM model.

The architecture of this model is illustrated below.

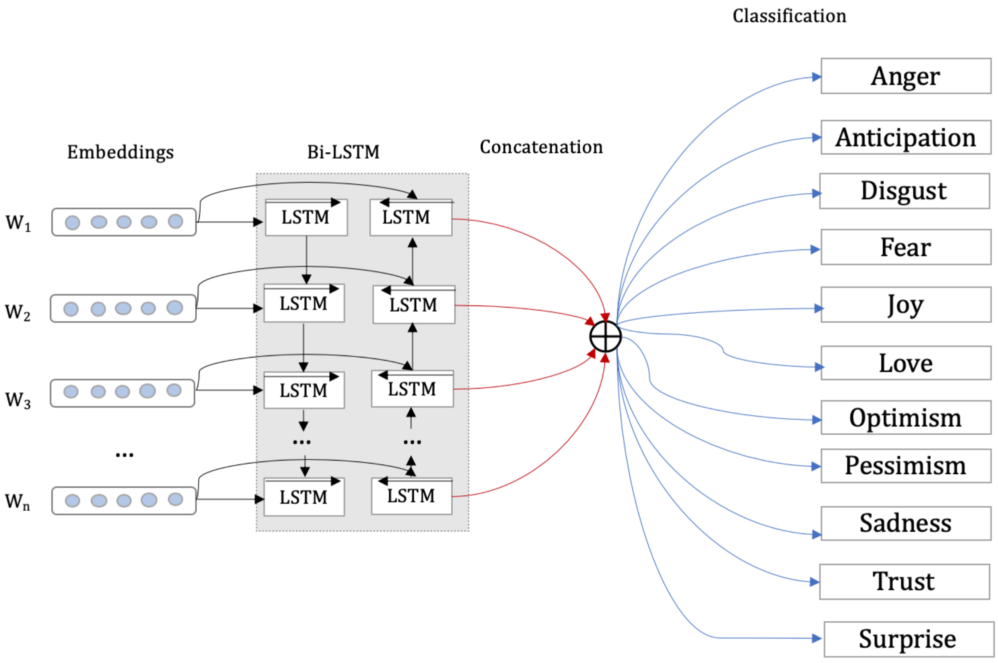


Figure 11 BiLSTM

Model 2: BiLSTM with single attention

In most cases, not all words in a tweet/text contribute evenly to the presentation of the tweet, so we take advantage of the word-attention mechanism to grab the words that are important to the emotion of tweet and sum the presentation of those instructive words to model a tweet vector. Formally,

$$U_{ij} = tanh(Wh_{ij} + b)$$

$$U_{ij} = tanh(Wh_{ij} + b)$$

$$\alpha_{ij} = \frac{exp(U_{ij}U_w)}{\sum_j exp(U_{ij}U_w)}$$

$$k_i = \sum_j (\alpha_{ij}h_{ij})$$

Figure 12 Attention formula

To obtain the context vector U_{ij}, we multiply weight W with the output of Bi-LSTM h_{ij} and add the bias value b. After this, as the similarity of U_{ij} with a word-level context-vector U_w, we measure the vital contribution of the word and obtain an importance weight α_{ij} via a SoftMax function. Then, we calculate the tweet vector k_i based on the weights as a weighted aggregation as described by (Zichao Yang, 2016). The Figure 3 is presenting Bi-LSTM model with single attention layer.

The architecture diagram of the model is illustrated below

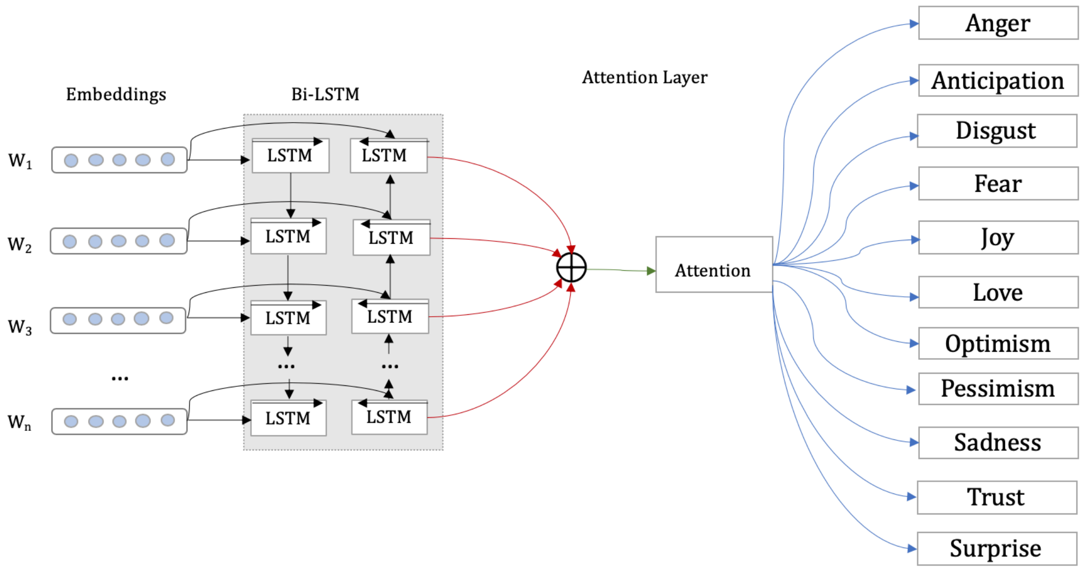
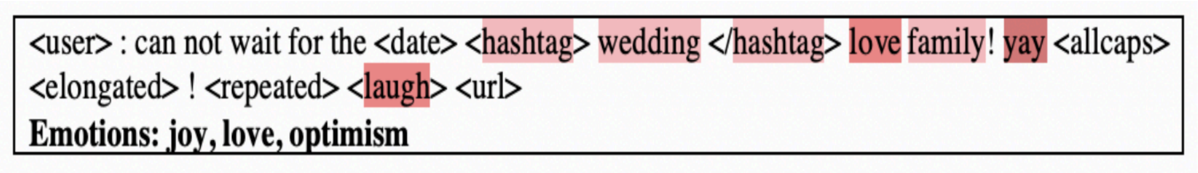


Figure 13 Bi-LSTM with single attention

Model 3: BiLSTM with multiple attentions

Till now, we a basic neural network architecture with attention frameworks, is in place. But still, the performance is not as good as the state-of-the-art. The work done till now is quite basic and can be called as the baseline of deep learning models with NLP. With this concern in mind, the following architecture was proposed and then tested on the dataset.

Instead of one attention layer, this model employed, 11 different attentions for 11 different emotions, respectively. In essence, in the code, there is an array of length:11, of attention weights and their respective biases. having this architecture in place, the model was learning important words for each type of emotion and saving the weights separately. This allowed the model to capture more information from the train set and was able to perform significantly better than the previously implemented architectures. The working of attention on a given input sequence can be better illustrated using the example below.



<user> : can not wait for the <date> <hashtag> wedding </hashtag> love family ! yay <allcaps> <elongated> ! <repeated> <laugh> <url>

Emotions: joy, love, optimism

Figure 14 Attention illustration

Now, in this model, there will be separate attention memories for every emotion present in the sentence. The architecture diagram of the model is illustrated below.

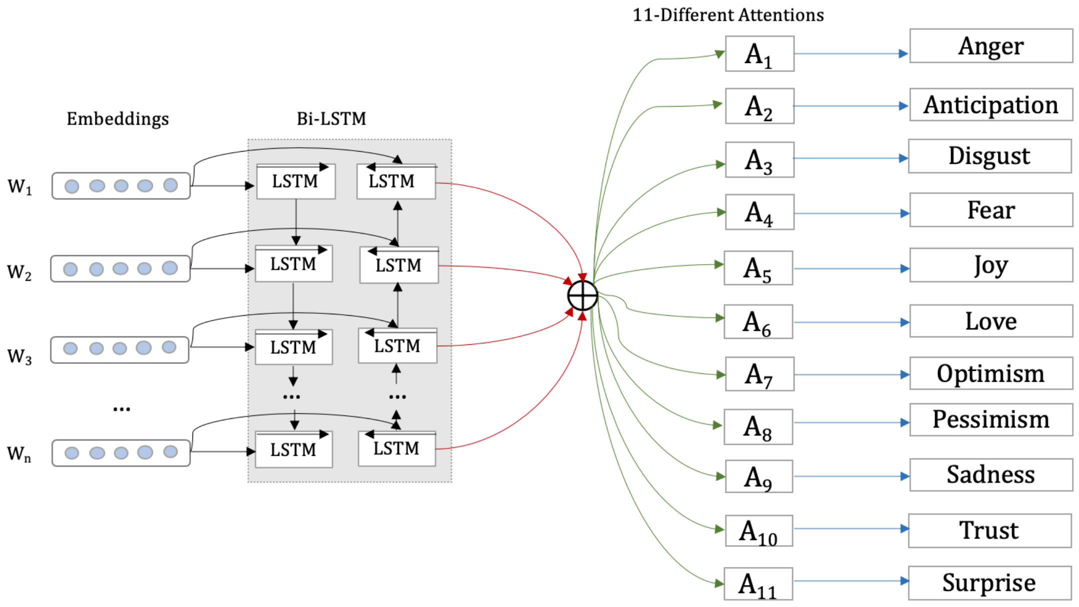


Figure 15 BiLSTM with multiple attentions

Model 4: Transfer learning (RoBERTa)

The final and the most complex model used for this problem is a version of BERT. BERT stands for Bidirectional Encoder Representations from Transformers. The version of BERT that is used in this thesis research is RoBERTa. RoBERTa stands for Robustly Optimized BERT pretraining Approach, presented by Facebook AI. According to (Yinhan Liu, 2019) BERT, is significantly undertrained and has the ability to outperform any future model that comes after it. Having said that, the RoBERTa version of BERT is robustly optimized for many NLP tasks and has the ability to outperform many similar models.

Using this architecture, the dataset was trained on it, as a multilabel classification problem. This training and testing then yield very promising and new state-of-the-art results on the problem. The following table represents out hyperparameter tuning for the RoBERTa architecture.

Parameter	Value
Fp16	True
fp16_opt_level	01
Maximum sequence length	128
Train batch size	8
Gradient accumulation steps	1
Evaluation batch size	8
Training epochs	3
Weight decay	0
Learning rate	4e-5
Adam epsilon	1e-8
Max grad. norm.	1.0

Table 21 Transfer learning hyper parameters

The following section contains a description about these hyperparameters

Fp16

Fp16 is set to be true, meaning that it will use half precision floating point arithmetic during the training computations. The benefit of using Fp16 is the significant enhancement in training times, without the undesirable loss in performance. More specifically, Fp16 will reduce the memory, by cutting the tensor size, in half; which will, in turn speed up the training time too as the weights are now half in size and their computation will not take much time (when compared to usual).

Fp16 opt level

The Fp16 opt level is set to 1. Fp16 introduces half precision floating point, while the opt level elaborates the optimization level for Fp16. Having an opt level “01”, means that the model will use mixed precision training implementations.

Maximum sequence length

The maximum sequence length is set to 128, meaning that after tokenization of the input document, the maximum length of it will be 128 tokens, which will then be fed into the model.

Trian batch size

This parameter represents the number of instances of input data that will be passed into the model for training in one iteration. Out batch size is 8, so 8 instances will go for training in one iteration.

Gradient accumulation steps

This parameter defines the number of steps to wait, before doing a backward pass on the model, or updating any of the variables. The gradient accumulation steps are set to be 1, which means that the model will do a backward pass after every forward pass.

Evaluation batch size

This parameter defines the size of the evaluation dataset batch. We have an evaluation batch size of 8.

Training epochs

Train epoch defines the number of times the model is trained on the input data completely. We have a train epoch size of 3.

Weight decay

Weight decay controls the weights from growing too large, by multiplying the weights by a value less than 1 after each update. We do not use weight decay in our model and hence have a parameter value of 0.

Learning rate

Learning rate defines the amount by which the weights are updated during the training process. This is usually a small positive value, and for our model, it is set to be 4e-5.

Adam epsilon

Neural networks use the Adam optimizer algorithm. The Adam epsilon parameter ensures that there is no possibility of a division by zero error in the implementation, and hence, has a very small positive value. The Adam epsilon value for our model is set to 1e-8

Max grad norm

This parameter is used to clip global grad. norm. It has a default value of 1, so in our case.

Chapter 5

Results

Introduction

In this section, the results from various implementations of the solution are described and discussed. This section will compare the results achieved with other implementations' results and will contain a discussion module to explain the findings. Two of the models proposed in thise research have surpassed the current state-of-the-art. The following section details the evaluation metrics and results in comparison to other published outcomes.

Experiments and results

In training the models, we use Adam optimizer (Ba., 2014) with auto mini-batching and 0.3 dropout size. For the development of our models, we use DyNet (Neubig et al., 2017) (Graham Neubig, 2017). We use the following conventions in results' Table 1. In the first column, Models means to the different developed models with specific settings. The" LSTM-DO-SA" stands for Bi-Directional Long Short-term Memory model with Dropout and Single-Attention mechanism, same for "Bi-LSTM- DO-MB-MA" but with Mini-Batching and Multiple Attentions. For the emotion analysis task, the evaluation metric is multi-label accuracy (Jaccard Index) although we have also reported the Macro F1 and Micro F1 scores. We can see that our Simple Transformer models (XLNet, DistilBERT, and RoBERTa) yield better results (Accuracy = 61.2) as compared to other models and outperformed the state-of-the-art results (57.4 - an official ranking of SemEval-2018 Task 1: E-c competition). Our purposed Multiple Attentions model also surpassed the state-of-the-art scores with a thin margin. On the multi-label emotion classification task, the simple transformers outperform other neural network models (LSTM, Bi-LSTM, Bi-LSTM-DO-SA, and Bi-LSTM-DO-MB-MA). This can be ascribed to the reality that our source corpus from SemEval-18 was helpful to boost the model classification performance when categorizing the tweets into none, one or more of the 11 emotion labels.

The following table shows the results achieved using each of the methodologies.

Model	Accuracy	Micro F1	Macro F1
LSTM	53.9	66.9	54.1
BiLSTM	55.1	65.0	50.1
BiLSTM – DO – SA	56.4	68.0	54.5
BiLSTM – DO – MB - MA	**58.1**	**70.1**	**56.9**
XLNet	59.4	69.9	58.1
DistilBERT	60.3	71.1	58.5
Best performing model			
RoBERTa	**61.2**	**73.7**	**59.9**
State of the art			
BiLSTM self-attention	57.4	69.7	57.4

Table 22 Result comparison

Discussion

Two of the approached described above have successfully crossed the current state-of-the-art results in multilabel emotion classification. One was using multiple-attention over a BiLSTM network, and the other included training a RoBERTa transfer learning model. The reason for multiple attention framework being better than the state-of-the-art, was the 11 distinct attentions, one for each emotion label. This layout gave the model more information about emotion triggering words and more attention was given to each of them. On the other hand, transfer learning, being the latest breakthrough in NLP, as expected outperformed everything. The model uses whole context of the document at once to make decisions about the respective emotion tags.

In this research, the effect of data quality on the performance of deep learning models is also examined. It is a fact that social media content, be it text, audio or video is very informal. When on the internet, people do not resort to the use of formal language and sentence structure, it is actually the complete opposite. People use short forms of words, that are not a part of standard English writing; they use distorted forms of words, elongations, the use of characters to make faces and much more. All these attributes of informal writing can be easily considered and eradicated as noise. But, the problem with this eradication is the loss of problem-critical information. For the problem researched in this thesis, these stylistic features and noise actually contribute to the triggering and detection of emotions in the text. So, it is not a good idea to lose this information and doing so will result in the degradation of performance and output quality. We did experiments on a "cleaned version" of the dataset, by removing all this noise and the difference was an accuracy loss of over 5%. This proves that the definition of quality data will vary from task to task. Somewhere, it will be absolutely necessary to get rid of this information, while on the other hand, there will be cases, like the problem discussed in this thesis, where the noise in the data will be actually telling more than the standard English words. Hence concluded that data quality plays an important role and the stylistic information should be preserved and converted into machine readable format for the better performance of the learning models.

The results are promising and set a new benchmark in multilabel emotion classification research area. The hyperparameter tuning of the model played a positive role in the achievement of a superior accuracy. These parameters are noted down and the trained models are saved for further use, and research.

Chapter 6

Conclusion and future work

Specific goals

The following specific goals are achieved

- Corpus cleaned and stylistic features extracted from the dataset that represent emotion triggering.
- Useless information removed from the documents that does not contribute to emotion cause but causes noise instead.
- Features extracted from the corpus, converted to word embeddings for learning by the AI models
- Designed an LSTM model with basic parameters.
- Designed a BiLSTM model with basic parameters
- Designed a BiLSTM model with single attention layer so that the word attentions are preserved
- Designed a BiLSTM model with multiple attention layers so each emotion gets an attention weight.
- Trained and tuned a transfer learning RoBERTa architecture for supreme performance
- Evaluated system performance with the current state-of-the-art. Set a new benchmark in this problem.

Final contributions

The final technical and scientific contributions are as follows:

- Creation of several deep learning models for the problem
- Training of several transfer learning models for the problem
- Extraction of stylistic and emotional features from the text
- Comparison of current results with state-of-the-art
- Setting a new benchmark for the research to follow
- A research article to be published on the findings and methodologies used.

Future work

The techniques employed have already produced benchmark results, but, as for any computer science problem, there is always a better way to do it. The future work in this area will include the creation and training of a novel transfer learning model that is optimized for multilabel classification. This problem is hard, given the 11 emotions and their binary nature. A dedicated transfer learning model, which is pre-trained on emotional tasks would be a promising addition to the research space. Marginal improvements can also be achieved by hyperparameter tuning of the transfer learning models. We were unable to test a lot of hyperparameter combinations, since a single training and testing cycle was taking days to run.

Bibliography

Mohammad, S. B.-M. (2018). SemEval-2018 Task 1: Affect in Tweets. *Proceedings of The 12th International Workshop on Semantic Evaluation.*

Pennington, J. S. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Ying Zeng, H. Y. (2016). A convolution bil-stm neural network model for chinese event extraction. *In Natural Language Understanding and Intelligent Applications, Springer*, 275–287.

Schmidhuber, S. H. (1997). Lstm can solve hard long-time lag problems. *In Advances in neural information processing systems*, 473–479.

Alex Graves, A.-r. M. (2013). Speech recognition with deep recurrent neural networks. *In 2013 IEEE international conference on acoustics, speech and signal processing,* (pp. 6645–6649). IEEE.

Huo, K. C. (2016). Training deep bidirectional lstm acoustic model for lvcsr by a context-sensitive-chunk bptt approach. IEEE/ACM Transactions on Audio, Speech and Language Processing.

Zichao Yang, D. Y. (2016). Hierarchical attention networks for document classification. . *In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, (pp. 1480 - 1489).

Yinhan Liu, M. O. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv. Ba., D. P. (2014). Adam: A method for stochastic optimization. . arXiv.

Graham Neubig, C. D. (2017). Dynet: The dynamic neural network toolkit. arXiv.

González, J. H., & Pla, F. (2018). ELiRF-UPV at SemEval-2018 Tasks 1 and 3: Affect and Irony Detection in Tweets. *Proceedings of The 12th International Workshop on Semantic Evaluation. .*

Zhang, Y. W., & Zhang, X. (2018). YNU-HPCC at SemEval-2018 Task 1: BiLSTM with Attention based Sentiment Analysis for Affect in Tweets. *Proceedings of The 12th International Workshop on Semantic Evaluation. .*

Abdullah, M. &., & Shaikh, S. (2018). TeamUNCC at SemEval-2018 Task 1: Emotion Detection in English and Arabic Tweets using Deep Learning. *Proceedings of The 12th International Workshop on Semantic Evaluation.*

Rozental, A. &., & Fleischer, D. (2018). Amobee at SemEval-2018 Task 1: GRU Neural Network with a CNN Attention Mechanism for Sentiment Classification. *Proceedings of The 12th International Workshop on Semantic Evaluation.*

Meisheri, H. &., & Dey, L. (2018). TCS Research at SemEval-2018 Task 1: Learning Robust Representations using Multi-Attention Architecture. *Proceedings of The 12th International Workshop on Semantic Evaluation. .*

Karasalo, M. N., & Bolin, U. W. (2018). FOI DSS at SemEval-2018 Task 1: Combining LSTM States, Embeddings, and Lexical Features for Affect Analysis. *Proceedings of The 12th International Workshop on Semantic Evaluation.*

Mulki, H. A., & Babaoglu, I. (2018). Tw-StAR at SemEval-2018 Task 1: Preprocessing Impact on Multi-label Emotion Classification. *Proceedings of The 12th International Workshop on Semantic Evaluation.*

Dragoni, M. (2018). NEUROSENT-PDI at SemEval-2018 Task 1: Leveraging a Multi-Domain Sentiment Model for Inferring Polarity in Micro-blog Text. *Proceedings of The 12th International Workshop on Semantic Evaluation.*

Dragoni, M. (2018). NEUROSENT-PDI at SemEval-2018 Task 1: Leveraging a Multi-Domain Sentiment Model for Inferring Polarity in Micro-blog Text. *Proceedings of The 12th International Workshop on Semantic Evaluation.*