EMOTION RECOGNITION FROM TEXT STORIES USING AN EMOTION EMBEDDING MODEL

SEMINAR REPORT

Submitted by

VISHNU B DEV

SBC17CS057

submitted to APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SREE BUDDHA COLLEGE OF ENGINEERING

ALAPPUZHA - 690 529
OCTOBER 2020

SREE BUDDHA COLLEGE OF ENGINEERING

PATTOOR P.O., ALAPPUZHA

BONAFIDE CERTIFICATE

This is to certify that the seminar report entitled **"Emotion Recognition from Text Stories Using an Emotion Embedding model "** submitted **by VISHNU B DEV, SBC17CS057** in partial fulfilment of the requirements for the award of the **Degree of Bachelor of Technology** in **Computer Science and Engineering** is a bonafide record of the work carried out under my guidance and supervision at college.

MS. LAKSHMI S	**MR. ARUN P S**	**DR. S.V. ANNLIN JEBA**
(Supervisor)	(Seminar Co-Ordinator)	Professor and Head
Assistant Professor	Assistant Professor	Dept. of CSE
Dept. of CSE	Dept. of CSE	

ACKNOWLEDGEMENT

I express my sincere gratitude to our respected principal **Dr. S. Suresh Babu** for his valuable support and advice.

I express my sincere thanks to **Dr. S.V. Annlin Jeba**, Head of the Department, Department of Computer Science and Engineering, Sree Buddha College of Engineering, Pattoor for her valuable guidance and suggestions throughout the entire seminar work.

I would like to thank my seminar coordinator, **Mr. Arun P S**, Assistant Professor, Department of Computer Science and Engineering, Sree Buddha College of Engineering, Pattoor for his guidance and advice during the work.

I would like to thank with deep sense of gratitude and obligation to my seminar supervisor, **Ms. Lakshmi S,** Assistant Professor, Department of Computer Science and Engineering, Sree Buddha College of Engineering, Pattoor for her valuable guidance and suggestions throughout the entire seminar work.

I express my sincere gratitude to all the members of the Department of Computer Science and Engineering, Sree Buddha College of Engineering, for their encouragement and valuable assistance.

In particular, I am thankful to all those who have helped me directly or indirectly in completing this work

.

Vishnu B Dev

ABSTRACT

The major role of artificial intelligence is to mimic the human traits and make the computer-human interactions less robotic. Emotion detection is a huge step towards making the computer respond with empathy. In this paper, the analysis of emotions in a story text is done using an emotion embedding model. Emotion embedding model here refers to an embedded layer trained in CNN emotional classification learning process. CNN is a class of deep neural networks which take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Various approaches in analysing emotions in a text used in this paper can be roughly divided into two categories: a keyword-based method and a learning-based method. The approach to textual emotion analysis here involves collecting Tweet data, building an emotion embedding model, extracting emotional words from text stories and textual story emotion recognition using an emotion embedding model. The method helped increasing the accuracy of results compared to predecessor methods.

Keywords: Artificial intelligence, Emotion Embedding Model, CNN

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CNN	**-**	**Convolutional Neural Network**
NLTK	**-**	**Natural Language Toolkit**
VADER	**-**	**Valence Aware Dictionary for Sentiment Reasoning**
ROC	**-**	**Commonsense Reasoning for Natural Language Understanding**
GloVe	**-**	**Global Vectors**

CHAPTER 1
INTRODUCTION

1.1 Overview

Recognizing user's emotions is a major challenge for both humans and machines. On one hand, people may not be able to recognize or state their own emotions at certain times. On the other hand, machines need to have accurate ground truth for emotion modelling, and also require advanced machine learning algorithms for developing the emotion models.

The research in this area gradually draws attentions because it provides a rich possibility in both theoretical and practical domains. It would be interesting to validate psychological theories via objective computer programs, and applications such as artificial pets may participate in blogs with the input of masters' emotion states from texts.

Definitions about emotion, its categories, and their influences have been an important research issue long before computers emerged, so that the emotional state of a person may be inferred under different situations. Since Picard proposed the concept of affective computing in 1997, the role of emotions in human-computer interactions has been gradually established, and this domain soon attracts interdisciplinary researchers from computer science, biology, psychology, cognitive science and so on. Following the trend, the computational research of emotion detection from texts emerged to determine human emotions from another point of view.

Methods of Emotion Detection

Keyword-based Method

Keyword-based methods are the most intuitive ways to detect textual emotions. It serves as the starting point of textual emotion recognition. while detecting emotions based on related keywords is very straightforward and easy to use, the key to increase accuracy falls to two of the preprocessing methods, which are sentence parsing to extract keywords, and the construction of emotional keyword dictionary. Parsers utilized in emotion detection are almost ready-made software packages, whereas their corresponding theories may differ from dependency grammar to theta role assignments. On the other hand, constructing emotional keyword dictionary would be naval to other fields. As this dictionary collects not only the keywords, but also the relations among them, this dictionary usually exists in the form of thesaurus, or even ontology, to contain relations more than similar and opposite ones.

Learning-based Method

Researchers using learning-based methods attempt to formulate the problem differently. The original problem that determining emotions from input texts has become how to classify the input texts into different emotions. Unlike keyword-based detection methods, learning-based methods try to detect emotions based on a previously trained classifier, which apply various theories of machine learning such as support vector machines and conditional random fields, to determine which emotion category should the input text belongs.

However, comparing the satisfactory results in multimodal emotion detection, the results of detection from texts drop considerably.

Difficulties in Determining Emotion Indicators The first problem is, though learning-based methods can automatically determine the probabilities between features and emotions, learning-based methods still need keywords, but just in the form of features. The most intuitive features may be emoticons, which can be seen as author’s emotion annotations in the texts. The cascading problems would be the same as those in keyword-based methods.

Nevertheless, lacking of efficient features other than emotion keywords, most learning-based methods can only classify sentences into two categories, which are positive and negative. Although the number of emotion labels depends on the emotion model applied, we would expect to refine more categories in practical systems.

Hybrid Method

Since keyword-based methods with thesaurus and naïve learning-based methods could not acquire satisfactory results, some systems use a hybrid approach by combining both or adding different components, which help to improve accuracy and refine the categories. In hybrid methods, emotions are detected by using a combination of emotional keywords and learning patterns collected from training datasets, in addition to information from different sciences, like human psychology. Few works addressed the problem of extracting emotions from text that does not contain emotional keywords.

1.2 Methodology

1.2.1 NLTK VADER

Our approach to textual emotion analysis is by building an emotion embedding model. The most emotional word in each story sentence is extracted using NLTK VADER sentiment analyser. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabelled text data.

VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text.

For example- Words like *'love', 'enjoy', 'happy', 'like'* all convey a positive sentiment.

Based on Plutchik's 8 basic emotion types (Anger, Anticipation, Disgust, Fear, Joy, Trust, Sadness, Surprise), emotional classification consisting of those 8 emotion words or relevant similar words is done.

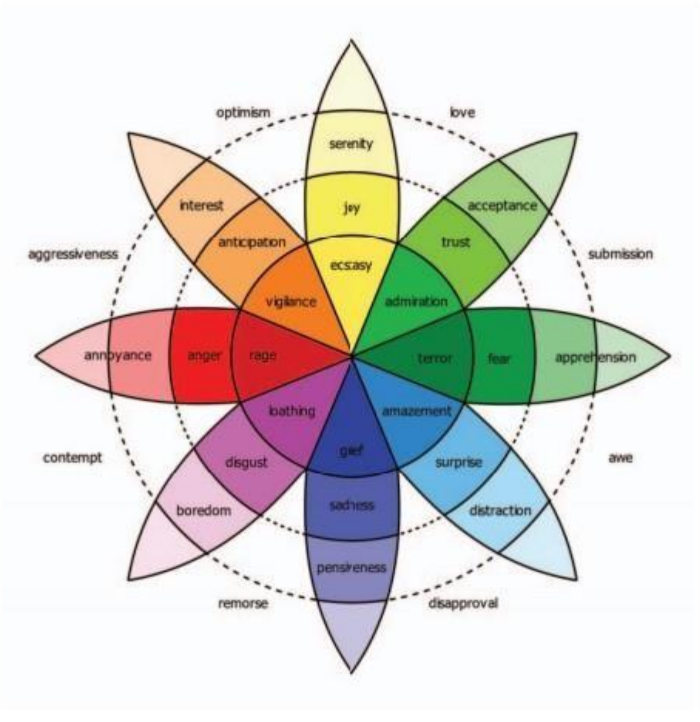

Fig 1.1. Plutchik's Wheel of Emotions

1.2.2 GloVe

GloVe, coined from Global Vectors, is a model for distributed word representation. The model is an unsupervised learning algorithm for obtaining vector representations for words. This is achieved by mapping words into a meaningful space where the distance between words is related to semantic similarity. Training is performed on aggregated global word-word cooccurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. It is developed as an open-source project at Stanford. As log-bilinear regression model for unsupervised learning of word representations, it combines the features of two model families, namely the global matrix factorization and local context window methods.

1.2.3 CNN

A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analysing visual imagery. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. layer. Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights.

In recent years, CNNs have been applied to natural language processing, as a model for classifying sentences. When a sentence is given as input to CNN, each word of the sentence is assigned to an input array. For each word, a corresponding word vector is taken from a pretrained word embedding model and a cluster vector is taken from a cluster model. Performance improvement can be obtained by adding semantic features of each word to CNN-based sentence classification model.

1.3 Scheme of the report

Chapter 1 gives an overview of the topic, objective, its applications etc. and the methodologies used for implementing the objective in detail.

Chapter 2 gives some related works done before.

Chapter 3 gives an overview of the system.

Chapter 4 tells about the advantages and disadvantages of the system.

Chapter 5 concludes the topic.

Chapter 6 gives the future work that can or may be done in the area.

CHAPTER 2
LITERATURE SURVEY

2.1 CNN based Sentence Classification with Semantic Features using Word Clustering

Classifying texts into one or more categories can be used for evaluating reviews as scores, classifying spams, or categorizing the subject of documents. It also plays a key role in fields such as understanding user's intention of speech, extracting and providing documents matching a topic, and classifying the language of a sentence in machine translation. Traditional approaches to sentence classification include statistical methods utilizing word frequency in documents and machine learning algorithms such as K Nearest Neighbours or Bayesian networks. Recently, deep learning based approach to text classification has been proposed.

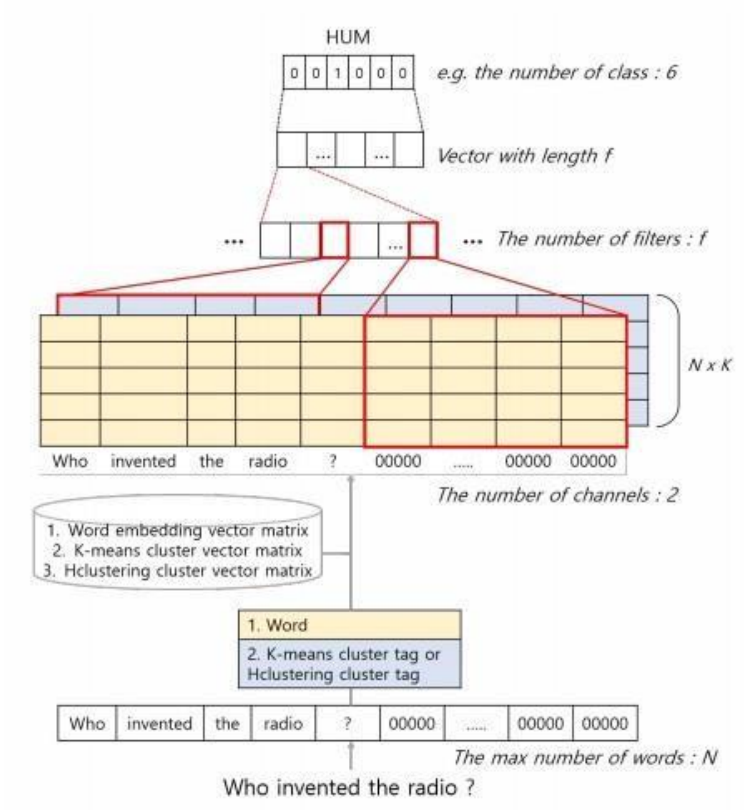

Fig 2.1. CNN-based sentence classification model with semantic features generated by word clustering method

Here combination of word vectors and their cluster information for CNN-based sentence classification is done. In order to do that, cluster information should be extracted by word clustering method such as K-means clustering and Hierarchical clustering. Since the extracted cluster information is cluster label information, it should be converted into a vector value. Thus, in the case of the K-means clustering, we use the centroid value (centre of gravity) of the cluster of a word, and in the case of the hierarchical clustering, we calculate the average of the word vectors included in a cluster.

When a sentence is given, each word of the sentence is assigned to an input array of size N (the number of words in the longest sentence in the dataset). For each word, a corresponding word vector is taken from a pre-trained word embedding model and a cluster vector is taken from a cluster model. The word vector becomes the first channel, and the cluster vector becomes additional channels of the input. The size of each channel is $N \times K$, N is the length of input array and K is the number of dimensions for word vector or cluster vector, respectively. The filter size of the convolutional layer is fixed to k columns, but can have various sizes in the rows, such as $2 \times k$ or $3 \times k$. If the number of filters is f, the output of the convolutional layer is represented by f vectors. Then, these outputs are passed to a fully connected softmax layer and results in the probability distribution over labels.

Performance improvement can be obtained by adding semantic features of each word to CNNbased sentence classification model. By doing so, the performance has improved by 0.6% in the major classification and 1.96% in the minor classification compared to the baseline. This is because the semantic features from word clustering have been organized such that they can well represent the characteristics of minor class.

Additional experiments are needed to adjust the number of word clusters according to the number of classes for sentence classification. It is also necessary to apply the proposed model to other domains to show its generality.

2.2 Towards Text-based Emotion Detection

Emotion detection is developed to provide cues for further human-computer interactions, so computers may stand as social actors to achieve more believable interactions. Besides emotion detection from texts, much work has been done to detect users' emotion states from multimodal sources such as audio, gestures, and eye gazes over the last decade. While multimodal interactions with computers have shown to be appealing, the most common way for people to interact with computer systems is still via texts. As web 2.0 emerges, more and more people have blogs to share their feelings with unspecified public. Though sharing pictures and videos has becoming popular, texts and blog articles still stand as an important role for expressing emotions. Compared to research of emotion detection in multimodal fields, emotion detection from text is still not mature and requires more improvements to be assembled as practical applications. These improvements include better understanding about newly evolved vocabularies, incorporation of psychological theories to infer emotion behind texts, utilization of contextual knowledge, developing more advanced emotion detection methods that allow more categories of emotions and inference.

The research in this area gradually draws attentions because it provides a rich possibility in both theoretical and practical domains. It would be interesting to validate psychological theories via objective computer programs, and applications such as artificial pets may participate in blogs with the input of masters' emotion states from texts.

Emotion detection is an important research field in affective computing over the last decade as much research has been done to detect facial, audio, and gestural emotions. The detected emotions would stand as important clues for advanced human-computer interactions. On the other hand, emotion detection from texts draws less attention. However, even in the era of web 2.0, text-based input is still the most common way for humans to interact with computers, and thus emotion detection from texts should be refocused as an important research issue in affective computing.

This paper surveyed existing research of emotion detection and reviewed the limitations to improve detection capabilities, describing a proposal of integrated system architecture. These improvements include identification of newly-evolved vocabularies, systematic emotion ontology based on OCC model as background knowledge, and collaborative method to detect multiple emotions in the form of case-based reasoning.

2.3 Emotion Recognition from Text Based on Automatically Generated Rules

Recognizing user's emotions is a major challenge for both humans and machines. On one hand, people may not be able to recognize or state their own emotions at certain times. On the other hand, machines need to have accurate ground truth for emotion modelling, and also require advanced machine learning algorithms for developing the emotion models. Hard sensing methods and soft sensing methods have been traditionally used to recognize user's emotions.

Aim is to recognize the six emotions suggested by Ekman: happiness, sadness, anger, fear disgust and surprise. We reduce the problem of emotion recognition or emotion detection from text to the problem of finding relations between the input sentence and the emotional content within it. Intuitively, finding these relations relies on discovering specific terms (emotional keywords, verbs, nouns, etc.) in the input sentence and other deeper inferences that are related to the meaning of the sentence. Once these terms and their relation to the meaning of the sentence are found, they can be generalized and considered as emotion recognition rules (ERRs).

This system follows a 2-tier client-server architecture. The client is responsible for getting the input sentence to be classified from the user (or the dataset to be classified), and sending it to the server for processing. On the server side, the generalization on the data is made and results are computed and sent back to the client.

This paper introduced a new approach for classifying emotions from textual data based on a fine-grained level. Our contribution lies in performing complex syntactic and semantic analysis of the sentence and using various ontologies such as Wordnet and ConceptNet in the process of emotion recognition. Syntactic and semantic analysis of the sentence makes our classifier context sensitive, while using WordNet and ConceptNet helps our classifier generalize the training set, which leads to better coverage of emotion rules.

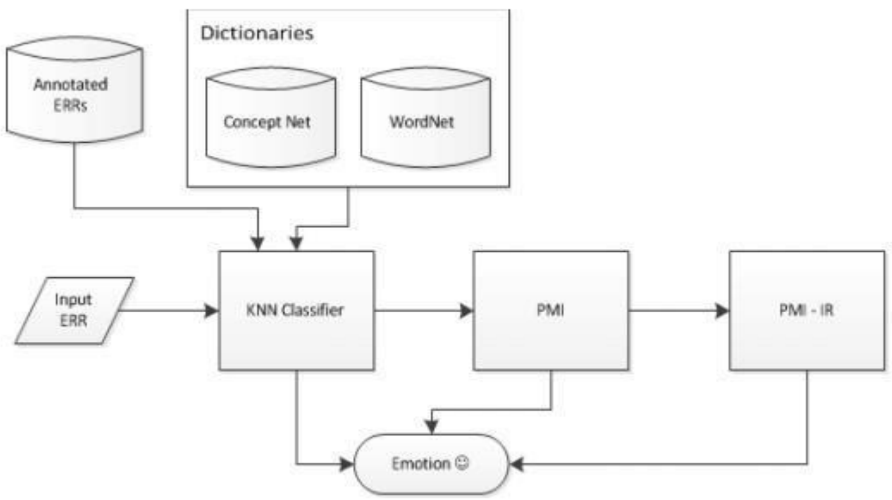

Fig 2.2. Classifier overview

This approach outperformed the state-of-the-art method in emotion classification from text (EmoHeart). Comparing the relations between the words of the sentence could lead to better accuracy than assigning an individual emotional rate for each word. We also showed that even with a training set different from the test set, our classifier performs better than EmoHeart. Moreover, the architecture of the proposed classifier is very flexible allowing it to be easily extended to classifying any number of emotions by providing a reasonably-sized training set that covers the required emotions.

CHAPTER 3
SYSTEM OVERVIEW

The goal is to build a hybrid model as an emotion embedding model for emotion detection from text. The model is a combination of keyword-based and learning-based approach where more accurate result can be achieved.

First, we collect emotional Tweet data with emotion hashtags based on the 8 basic emotion types. Second, using the emotional Tweet data, we build an emotion embedding model. Third, from a story dataset, the most emotional word in each story sentence is extracted using NLTK VADER sentiment analyser. Finally, based on the extracted word, we detect emotions using our emotion embedding model which are trained with Tweet emotion data.

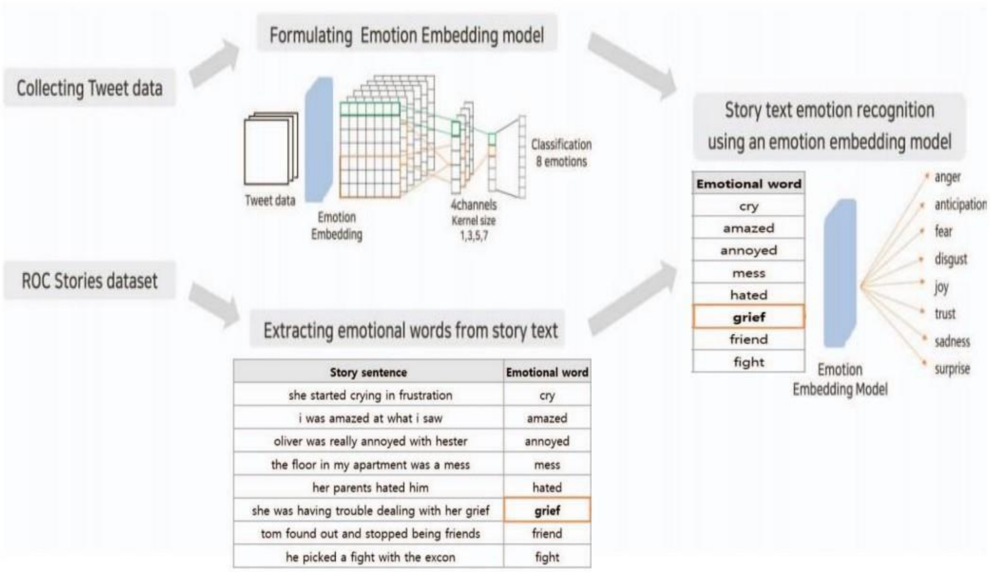

Fig 3.1. Emotion Embedding Approach to Textual Story Emotion Classification

3.1 Collecting tweet data

To build an emotion embedding model, a large text dataset with annotated emotions are needed. We employed 144,701 Tweet data, where each tweet was annotated with an emotion hashtag.

For example, a Tweet, “I broke up with my girlfriend. #sad” in the training data, “I broke up with my girlfriend” is the declarative sentence part, while #sad is an emotion annotation. Based on Plutchik’s 8 basic emotion types (Anger, Anticipation, Disgust, Fear, Joy, Trust, Sadness, Surprise), we collected Tweets with emotional hashtags consisting of those 8 emotion words or relevant similar words.

Tweetdata	
Emotion	Count
Anger	44488
Anticipation	8089
Disgust	8678
Fear	20012
Joy	22489
Trust	10697
Sadness	20462
Surprise	9786
Total	144701

Table 3.1. Count of Tweet Data Emotions

3.2 Building an emotion Embedding Model

In this step, a text emotion classification of Tweet data using the CNN learning algorithm is conducted. It is a supervised learning that trains the Tweet data corresponding to 8 emotions and validates the model performance by classifying emotions of the test data. In this learning process, an embedding layer to vectorize text is required. We employ GloVe as the initial embedding model. Then, through the backpropagation process, values of neural network layers are adjusted for emotion classification. The embedding layer is extracted when the best classification performance is reached. The extracted embedding layer has different values compared to the initial embedding layer. In other words, this layer is optimized to detect and classify emotions. We define this layer as our Emotion Embedding layer and apply this model to the emotion detection of text stories.

3.3 Extracting emotional words from text stories

For text stories, we employ the ROCStories dataset, which include 52,666 stories, each story consisting of 5 simple sentences (263,330 sentences in total). NLTK VADER Sentiment Analyzer is applied for detecting emotional words in the story sentences. The VADER sentiment analyser returns positive, negative, neutral and compound scores. The compound score value is an aggregated score of the other three scores- the sum of all the lexicon ratings which have been normalized between -1 and +1: -1 means the highest negativity; +1 means the highest positivity. Based on the sentiment polarities of words in the sentences, we select a word with the highest absolute value as a representative emotional word in the given sentence.

For example, in a sentence, “one day a guest made him very angry”, the polarity of each word is obtained by NLTK VADER Sentiment Analyzer as ([0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.121, 0.5106]). Here, the word “angry” is selected as the emotional word of the sentence with the highest polarity.

3.4 Textual Story Emotion Recognition Using an Emotion Embedding Model

As the last phase of our approach, we extract the emotions from text stories using our emotion embedding model. To determine the emotion of each sentence in the stories, we compute the cosine similarity between the selected emotional words using the NLTK VADER sentiment analyser and emotional hashtags for emotion annotation in Tweet data.

For example, in the story sentence “she was having trouble dealing with her grief”, the emotional word extracted from the analyser is ‘grief’.

Based on the cosine similarity values, similar words to ‘grief’ in the emotion embedding model were [’grief’, ’sadness’, ’sorrow’ and despair’]. Thus, a story sentence that includes the word ’grief’ is classified as the ‘Sadness’ emotion type.

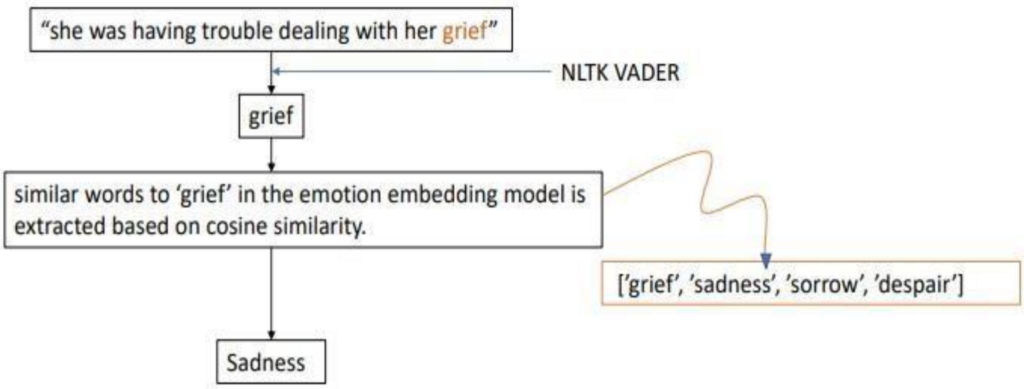

Fig 3.2. Emotion Extraction by computing cosine similarity

3.5 Evaluation result

A total of 137,052 story text sentences were analysed. The classification results show that Joy(22%), Sadness(19.88%), Fear(16.4%) and Anger(14.4%) are the top 4 emotions, occupying a majority of 73.55% of the total counts. These 4 emotions showed higher proportion because the Tweet data used in creating the embedding model also had a lot of Joy, Sadness, Fear, and Anger data (74.26%).

Result		
Emotion	Count	Percent
Anger	19711	14.4
Anticipation	13024	9.5
Disgust	3251	2.37
Fear	22480	16.4
Joy	31371	22.89
Trust	6769	4.94
Sadness	27245	19.88
Surprise	13201	9.63

Table 3.2. Emotion analysis result of story text

Among the well-classified examples, the sentence “I was amazed at what I saw” was classified as the emotion of Surprise, because the word ‘amazed’ was extracted as an emotional word. And, in the sentence “Oliver was really annoyed with Hester”, the word ‘annoyed’ was selected as an emotional word, so the emotion of the sentence was classified as Anger, which included ‘annoyed’.

On the other hand, a manual inspection of the results also reveals some mis-classified examples. The sentence “He was determined to play but knew he had to study at night” was classified as the emotion of Joy: the word ‘play’ was selected as the emotional word of the sentence, but the context showed that the emotion of the sentence was not Joy. Likewise, in case of a sentence containing negativity such as ‘not happy’, ‘happy’ was extracted as the emotional word: the sentence was classified as Joy, even though it is the opposite of the actual emotion of the sentence.

story sentence	emotional word	emotion class
she started crying in frustration	cry	sadness
i was amazed at what i saw	amazed	surprise
oliver was really annoyed with hester	annoyed	anger
the floor in my apartment was a mess	mess	disgust
her parents hated him	hated	disgust
she was having trouble dealing with her grief	grief	sadness
tom found out and stopped being friends	friend	sadness
he picked a fight with the excon	fight	anger
she felt prepared	prepared	trust
he decided to play but knew he had to study at night	**play**	**joy**
he was not happy about having to go to school	**happy**	**joy**

Table 3.3. Calculating Similarity between Story emotional words and emotion hashtags

3.6 Performance Evaluation

To evaluate the emotional embedding performance, 120 sentences (15 sentences for each of the eight emotions) of the story sentences were randomly selected. Four human raters evaluated the emotions of the sample sentences. All four raters are college students in their 20s: two males and two females. The raters determined the sentiment polarity of the given sentences and the emotion it belongs to. Based on the evaluation of human raters, we calculated the accuracy of each emotion. The Joy emotion showed the highest accuracy, while Anger results in the lowest accuracy. Inspection of some Anger labelled sentences suggests that these sentences often accompanied other negative emotions such as Sadness and Fear. On the other hand, sentences labelled as Joy did not accompany any other positive emotions.

CHAPTER 4
ADVANTAGES AND DISADVANTAGES

4.1 Advantages

- Able to detect the emotion of a particular sentence.
- More accuracy than keyword-based.
- More accurate compared to learning-based method.
- Produce accurate result for sentences without negative words.

4.2 Disadvantages

- Fails to identify contextual emotion from more than one sentence.
- Fails to identify emotions from sentences with negating expressions.

CHAPTER 5
CONCLUSION

This study presents a method to extract the emotion of a sentence using an emotion embedding model. First, an emotion embedding word model is built using the collected Tweet data annotated with hashtags. Then extracted the representative emotional word in each sentence of the ROC story data. The representative emotional word is then used to classify the emotion of the sentence leveraging the cosine similarity. Experiments are conducted and the results show that this approach is promising. This method analyses emotions from story texts based on the emotional words representing each story sentence. Therefore, this approach does not take into account the contextual information that can span multiple sentences. It also does not handle expressions negating the sentence such as 'no', 'little' , or 'not'. In future research, this emotion embedding model can be enhanced to detect contextual emotional information in the story text.

CHAPTER 6
FUTURE WORK

The proposed method can be upgraded in future researches to detect contextual emotional information in the story text. In that case, the model can be used in various fields of mental health care, customer satisfaction analysis, intelligent humanoids, etc.

REFERENCES

[1] Edward Chao-Chun Kao, Ting-Hao Yang, Chang-Tai Hsieh, Von-Wun Soo (2009). Towards Text-based Emotion Detection: A Survey and Possible Improvements. 2009 International Conference on Information Management and Engineering.

[2] Shadi Shaheen, Wassim El-Hajj, Hazem Hajj & Shady Elbassuoni (2014). Emotion Recognition from Text Based on Automatically Generated Rules. 2014 IEEE International Conference on Data Mining Workshop

[3] Armin Seyeditabari, Narges Tabari & Wlodek Zadrozny (2018). Emotion Detection in Text: a Review. arXiv:1806.00674v1 (cs)(2018).Cornell University.

[4] David B. Bracewell (2008). Semi-Automatic Creation of an Emotion Dictionary Using WordNet and its Evaluation.IEEE 2008 (CIS).

[5] Khodijah Hulliyah, Normi Sham Awang Abu Bakar & Amelia Ritahani Ismail (2017). Emotion recognition and brain mapping for sentiment analysis: A review. 2017 Second International Conference on Informatics and Computing (ICIC).

[6] Lingling Song, Yazhou Zhang & Yuexian Hou (2018). Convolutional Neural Network with Pair-Wise Pure Dependence for Sentence Classification. 2018 International Conference on Artificial Intelligence and Big Data

[7] Jiachen Du, Lin Gui, Yulan He & Ruifeng Xu (2017). A Convolutional Attentional Neural Network for Sentiment Classification.2017 International Conference on Security,Pattern Analysis and Cybernetics(SPAC).

[8] Hwa-Yeon Kim, Jinsu Lee, Na Young Yeo, Marcella Astrid, Seung-Ik Lee & YoungKil Kim (2018). CNN based Sentence Classification with Semantic Features using Word Clustering.IEEE 2018