S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

2019

Project Title	Sentiment Analysis on Twitter Corpus using Capsule Network
Student Name	P.D. Manusha Dilan
Registration No. & Index No.	2016/mcs/027 16440272
Supervisor's Name	Dr. D.A.S. Atukorale

For Office Use ONLY

Sentiment Analysis on Twitter Corpus using Capsule Network

P.D.M.Dilan

2019

Sentiment Analysis on Twitter Corpus using Capsule Network

A dissertation submitted for the Degree of Master of Computer Science

P.D.M. DILAN

University of Colombo School of Computing

2019

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: P.D.M. Dilan

Registration Number: 2016/mcs/027

Index Number: 16440272

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

____

Signature: Date:

Abstract

Twitter is one of the popular social media among people in recent years. Most people use it as an express opinion and view of their for regarding situation or matter. Because of this Twitter has significant value and interest on opinion mining and business and product marketing peoples. To achieve this goal sentiment analysis has used. Machine learning has recently gained success and popularity for sentiment analysis. So many machine learning approaches have emerged for sentiment analysis and got success. The newly emerging concept of deep learning is capsule network. This approach is emerging from convolutional neural networks. And it has overcome many problems that have carried with a convolutional neural network. Capsule network has considered as a new era of deep learning. This paper proposes a capsule network model to do sentiment analysis on Twitter data. And this model is compared with other few deep learning models to get better oversight on the proposed model. Purpose of this paper is to expose the capsule network model to do sentiment analysis on Twitter data and show the results and behaviors of the capsule network on sentiment analysis.

Keywords: Capsule network, Deep learning, Sentiment analysis, Twitter

Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisor Dr. D.A.S.Atukorale for the continuous support of my research, for his patience, motivation, enthusiasm, and immense knowledge in the past year. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my research project.

In addition, I am very grateful for the support of academic and management staff of University of Colombo School of Computing, Sri Lanka. And special thanks go to my manager of the company who gave me the support to balance the work life and academic work through the past year.

Finally, I must express my very profound gratitude to my parents and to my two brothers for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Background

Twitter is a microblogging system that has become popular among internet users[1]. In Twitter people can communicate via short message called tweets. Tweet message has limited to 140 characters. Also, it can include images, videos, URLs and hashtags. People can easily access Twitter through the web interface, mobile device application and short message service (SMS). Users can subscribe to other user's tweets and this called as following. These tweets are publicly visible by default and sender can restrict it to see only by followers.

Sentiment analysis is a process that uses to identify the subjectivity, polarity of text[1]. Polarity is emotions expressed in the sentence. Polarity normally represented in positive to negative scale. Middle ground considered as neutral. Subjectivity is expressing personal feelings, view and beliefs on sentence[2]. There are three levels of sentiment analysis.

- Document level

  Consider positivity or negativity that document is focusing on a single subject. This approach has the drawback that it considering that no fundamental difference between document and sentence level[2].

- Sentence level

  Consider that each sentence has a positive, neutral or negative sentiment. Although the sentence can be either positive or negative subjectify, sentiment may not be.

- Aspect level

  In this level, it will consider at word level opinion, which is either positive or negative.

Sentiment analysis in microblogging is different from traditional approaches because it forces the users to change their writing styles to a limited amount of characters. Also, microblogs are rich with images, hashtags, emoticons, user mentions, videos, and URLs. In this context, traditional approaches that developed for longer documents types are not suitable [1]. In twitter data as, traditional tasks it can detect the subjectivity and polarity at the message level, because of limited character size. Here also it can include figurative language like irony and sarcasm which will increase the difficulty of the analysis[1].

Sentiment analysis can be efferently used in social studies and marketing. For example, product manufacturer can know about, what people think about their product and how positive about the product and what kind of product that users are expecting[3]. The size of the twitter data and speed of increasing the size has attracted the huge range of application domains, from industry to politics.

Many types of research have conducted on sentiment analysis on twitter corpus using neural networks models[4], [5]. Though they have achieved the objective of sentiment analysis through these models, still analysis can be improved in sensitivity, accuracy, efficiency, performance and other aspects. So, analysis results can be greatly improved.

Capsule networks are newly emerged neural network technology[6]. In convolutional neural networks (CNN), accumulating a set of features at each layer but all spatial relationship information's are lost. But in capsule networks, it has the ability to get spatial relationship among data. In traditional CNN we can care only that whether model predicts the right classification or not. But with capsule network reconstruction can also be possible[6]. Because of these abilities capsule network are more powerful than other conventional neural networks.

A capsule is a set of neurons individually active for various properties of the type of objects, such as potions, size, and hue. Capsules are working independently. So when multiple capsules is use probability of much correct detection is high[7]. Capsule are routed by agreements. One capsule can give output to the next capsule according to the ability to predict the output. For each potential parent, capsule network can increase or decrease the connection strength[8]. In capsule networks learning is supervised.

Capsule structure will give more capacity to model sentiment. And knowledge in linguistic like lexicon, negation words need to be carefully incorporate in to model to get the best prediction accuracy[9]. For each category in sentiment, capsules can be created separately. Then using routing between these capsules, it can get the best results of the sentiment classification. Linguistic knowledge should be carefully incorporated into models to realize the best potential prediction of accuracy.

In this research we will be using capsule network for sentiment analysis, that will be carried out against English language twitter corpus only. Sentiment analysis will have done at the word level and message level. Emoticons also considered for sentiment analysis. Other languages rather than English will be discarded, and Image data also be not considered for analysis.

1.2 Motivation

In the new architecture of neural network called capsule introduced and it is very exciting to try out the new area. The popularity of Twitter in social media concepts has increased and it is very interesting to working in growing technology. Sentiment analysis can lead us to grasp the feeling and thoughts of people and community and it leads to the harmony of the society.

1.3 Statement of the problem

Sentiment analysis has used many areas in modern worlds like financial, political, marketing and many others. For example, analyzing customer feedback on a product, movie reviews, political opinion on people about political parties and so on. Doing a sentiment analysis is a difficult task because of the following reasons. One of the reasons is language usage. In one sentence may have more than one sentiment that makes hard to analyses which category it will fall. Another reason is sarcasm. For sarcasm, people may identify easily but for neural network, it is very difficult to understand, and classification will easily fail on this kind of sentences. Another one is the local dialects that people use. So, it is hard to generalize the approach on this kind of scenarios and network may try to fail.

Twitter data is a great asset for getting an opinion on many things like political, social behaviors, product reviews, movie reviews, how the share market will be affected by shares of a product. In Twitter data itself has challenges like hashtags, images, videos and emoticons that have a great effect on sentiment but analysis them is a hard task.

As shown in previous chapters' literature many researchers have tried to classify the Twitters on sentiment basis. All of them have achieved it on many levels and, have drawbacks on each of them.

Hypothesizing that, using the capsule network can improve the accuracy and performance of classification sentiment analysis on twitter corpus, then the other approaches.

1.4 Aims and Objectives

By using capsule networks, we can give better and more sensitive, accurate and efficient classification for twitter sentiments rather than traditional and other conventional neural network approaches and traditional sentiment analysis methods.

Optimizing the sentiment analysis process by using the capsule network for more stable and reliable output.

Comparing the capsule network model with other models and get more insight into them.

Chapter 2: Background/Literature Review

Aliaksei[4] and others have developed deep convolutional neural networks for Twitter sentiment analysis. In this approach, they have introduced a new model for initializing the parameter weights for the convolutional neural network. So that they can avoid the need to inject any additional features to model, that is crucial for the accuracy of the model. They have used unsupervised neural language model to train initial word embedding and then tuned by their deep neural model on a distantly supervised corpus. They are addressing the issue of providing the network with good initialization parameters that having a significant impact on the accuracy of the trained model. To overcome this, they are proposing a three-step process to train deep learning model for sentiment analysis. In this process using neural model word embedding are initialized and train on a large unsupervised collection of tweets. Then using the convolutional neural model to further improve the embedding on the large distant supervised corpus. The word embeddings and other parameters of the network obtained at the previous stage are used to initialize the network with the same architecture, which is then trained on a supervised corpus.

On this deep learning approach of sentiment analysis have done on tweets for predicting polarities of both message and phase levels. Their solution has succeeded in combining two traditional aspects such as unsupervised learning of text representations and learning on weakly supervised data.

Ghiassi[5] and others are using an approach of supervised feature reduction using n-grams and statistical analysis to develop a Twitter-specific lexicon for sentiment analysis. They are showing that reduced lexicon set, reduce modeling complexity, maintain a high degree of coverage over twitter corpus and yields improve sentiment classification accuracy. They developed comparable sentiment classification models using SVM. Then they have developed and show sentiment classification models using the Twitter-specific lexicon and the DAN2 machine learning approach. Then they compared and show (Figure 2.1) that DAN2 produces more accurate sentiment classification results than SVM while using the same Twitter-specific lexicon.

Accuracy by category.

Category	DAN2 train (%)	DAN2 test (%)	SVM test (%)	SVM-of test (%)
Strongly positive	83.9	69.7†	71.3**	67.4
Mildly positive	85.0	66.7†††	63.9	64.0
Mildly negative	92.5	89.9†	88.3	87.3
Strongly negative	95.7	95.1	94.6***	93.7

Figure 2.1 Table DAN2 vs SVM

In their approach they are stating that sentiment analysis has increased sensitivity, accounting for tweets with the mild sentiment, resulting in more accurate identification of the neutral category. And they find that emoticons to have high explanatory power rather than removing or discarding them. They have also shown that the DAN2 model was much better at finding the messages of interest.

Wang[9] and others have proposed a capsule model based on Recurrent Neural Network (RNN) for sentiment analysis. They built one capsule for each sentiment category (Figure 2). This each capsule has an attribute, a state, and three modules. Those are the representation module, probability module, and reconstruction module. Then attribute capsule has assigned to sentiment class. They have used these capsules on Movie Review dataset and Hospital Feedback dataset, without using any linguistic knowledge, and these capsules have outputting words with sentiment tendencies reflecting capsules' attributes.

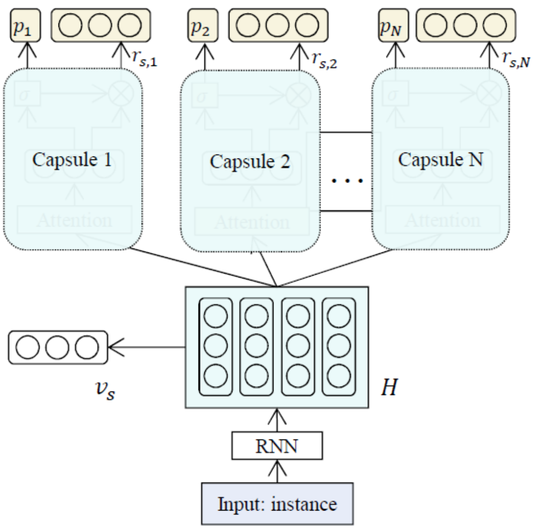

Figure 2.2 Architecture of RNN-Capsule

They state that their simple capsule model achieves state-of-the-art sentiment classification accuracy without any carefully designed instance representations or linguistic knowledge and capsule can output the words best reflecting the sentiment category.

Joshi and Deshpande[10] attempted to conduct different machine learning algorithms such that Naive Bayes and Maximum Entropy algorithms on Twitter sentiment analysis. They have preprocessed the twitter data by replacing URL, emoticons and user mentions with tags. Also, they removed hashtags and retweets from the data set. Then they have created a frequency distribution of the unigrams and bigrams present in the dataset and choose top N unigrams and bigrams for analysis. After that, they have fed data to the algorithm like Naive Bayes, Maximum Entropy or SVM. Their model can be improved to handle the range of sentiment. Also, they have discarded most of the symbols like commas, full-stops, an exclamation mark in preprocessing. These symbols may be helpful in assigning sentiment to a sentence.

Tang[11] and others have present a method that learns sentiment specific word embedding (**SSWE**) for Twitter sentiment classification. This method addresses the problem of sentiment analysis as it usually maps words with similar syntactic context but opposite sentiment polarity. They have developed three neural networks to effectively incorporate the supervision from sentiment polarity of text in their loss functions (Figure 3).

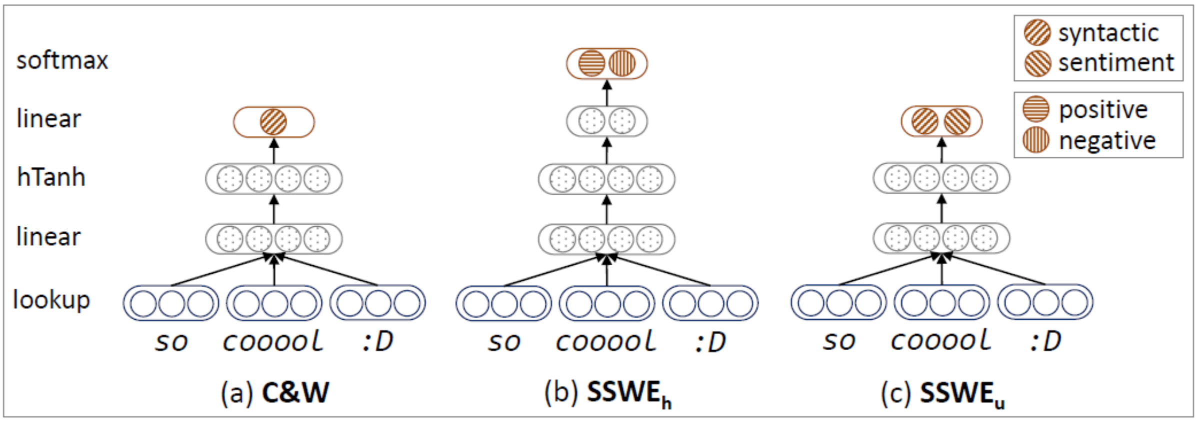

Figure 2.3 The traditional C&W model and Tang and other neural networks

They have trained SSWE with massive distant-supervised tweets selected by positive and negative emoticons. They state that a unified model combining syntactic context of words and sentiment information of sentences yields the best performance in experiments.

Fouad[12] and others have used the model to represent the input labeled tweets in the training phase using different features sets. In their model, the classification phase, the classifier ensemble is presented with different base classifiers for more accurate results. They also prune the irrelevant and insignificant features, using Information Gain feature selection technique. They have implemented a majority voting ensemble classifier with SVM, NB and LR as base learners which of algorithms are commonly used and have great success in the text classification problems. They have noticed that the reported results of the lexicon-based features and POS features enhanced the accuracy of the classifiers. As a drawback of this method, it not considering the neutral tweets when classification is done. Also, this model is not supported by other languages rather than English.

Pak[3] and others have collected the Twitter corpus and performed a linguistic analysis on them. Based upon it they built a classifier and output was compared with previous sentiment results. Their classifier is based on the multinomial Naive Bayes classifier that uses N-gram and POS-tags as features. They have used the presence of an n-gram as a binary feature, while for general information retrieval purposes, the frequency of a keyword's occurrence is a more suitable feature, since the overall sentiment may not necessarily be indicated through the repeated use of keywords. To increase the accuracy of the classification, they have discarded common n-grams, n-grams that do not strongly indicate any sentiment nor indicate objectivity of a sentence. From the observations, they conclude that authors use syntactic structures to describe emotions or state facts. Some POS-tags may be strong indicators of emotional text. This approach also not support other languages rather than English.

Barbosa[13] and Feng have done robust sentiment detection on Twitter from biased and noisy data. They have proposed an approach to automatically detect sentiments on tweets that explores some characteristics of how tweets are written and meta-information of the words that compose these messages. And they leverage sources of noisy labels as training data. Their approach creates a more abstract representation of tweets, instead of using a raw word representation of them, and although noisy and biased, the data sources provide labels of reasonable quality and, since they have a different bias, combining them also had brought some benefits. The main limitation of the approach is the cases of sentences that contain antagonistic sentiments.

Chapter 3: Methodology

Sentiment analysis will be carried out against the Twitter corpus dataset that available on the Kaggle website for research purposes.[14] This corpus contains 1,600,000 tweets extracted using the Twitter API. The Tweets have been annotated and can be used to detect the sentiment.

Figure 3.1 Word cloud of twitter corpus

In the word cloud (Figure 3.1) gives the idea about the frequency of the words that are the Twitter corpus dataset.

In the dataset, most of the sentences contain the words between 5 and 15. And sentiment distribution is not unequal, and it is carrying a fair weight in the dataset (Figure 5). There are no sentences contain more than 35 words. Top frequency of the words is about above 175000. In Figure 6 it shows the top most frequent 25 words and their word count. “I” is the most frequent word in the Twitter corpus dataset.

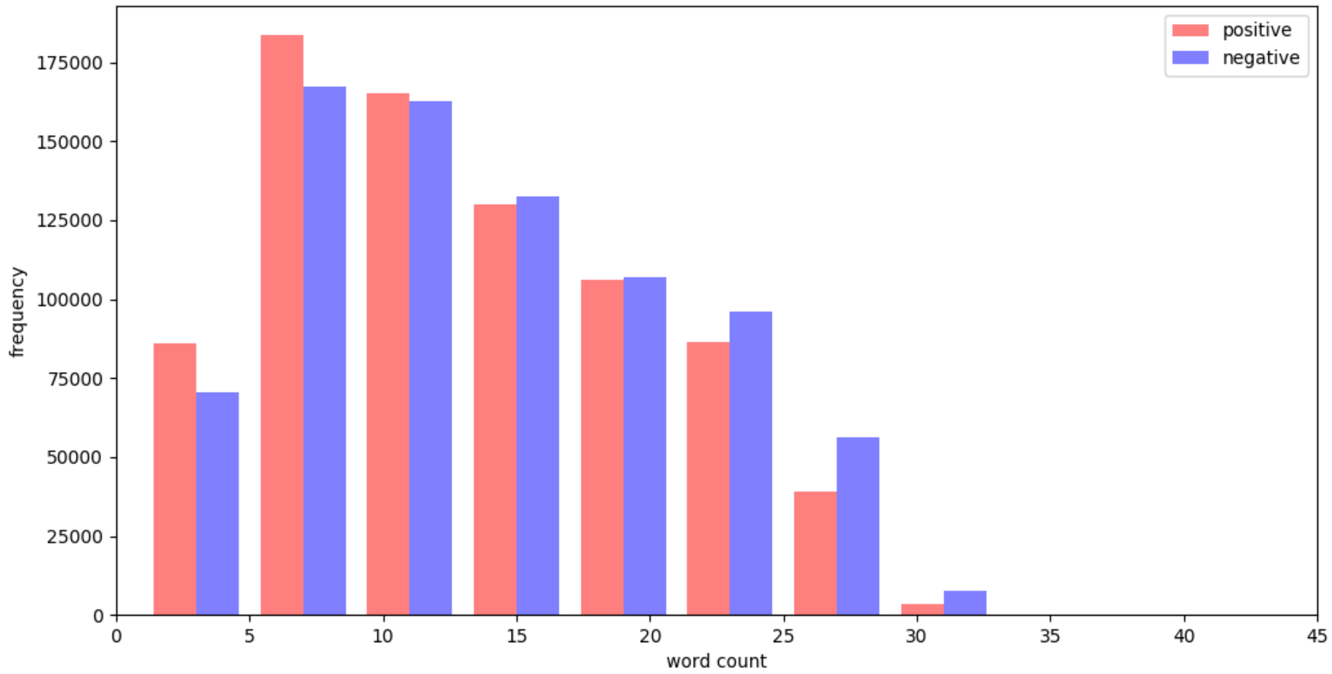

Figure 3.2 Word count vs word frequency of twitter corpus

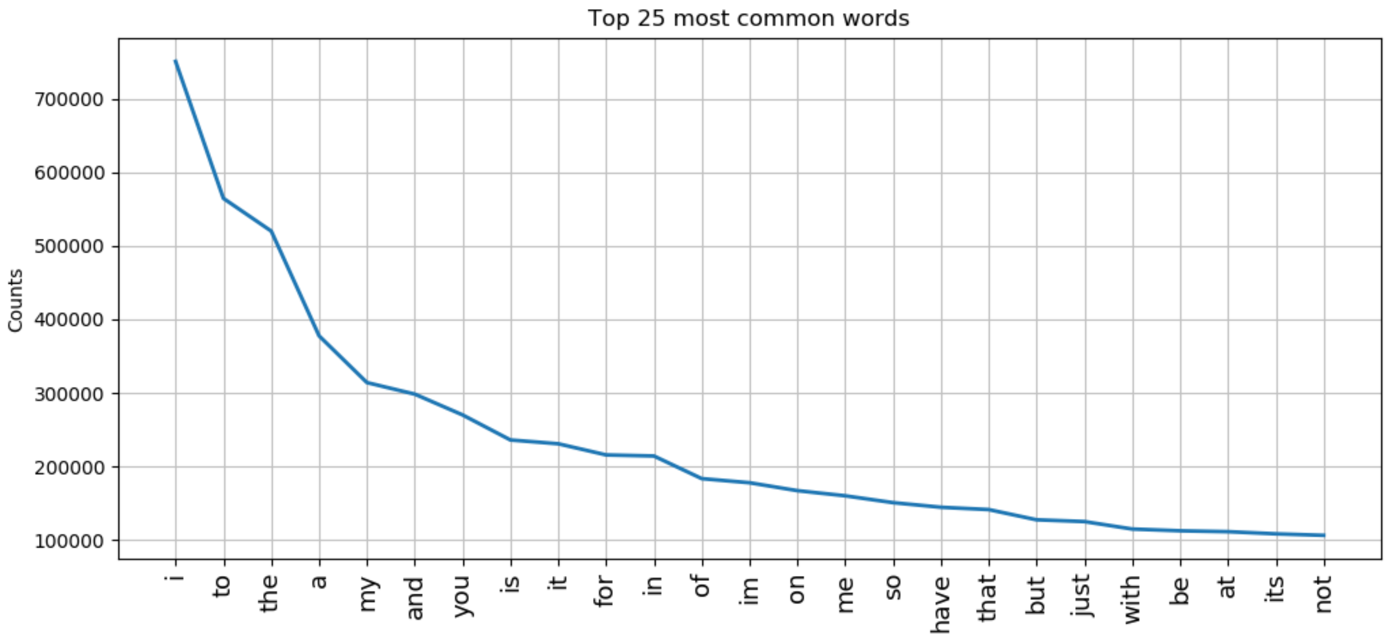

Figure 3.3 Top 25 most common words of twitter corpus

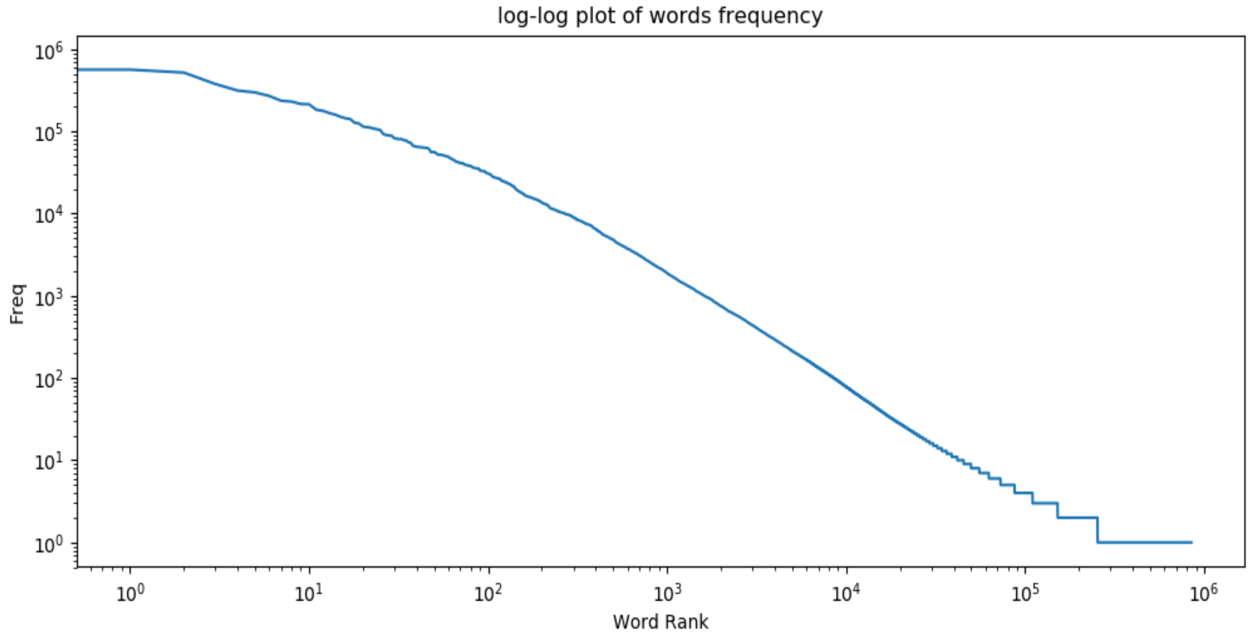

Figure 3.4 Words frequency of twitter corpus

Capsule network has several layers. Capsules in lower level correspond to simple entities and high-level capsules corresponding to that entity sends feedback to these low-level capsules.

Prediction vector by the capsules is $\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i$ **W_{ij}** is the weight matrix.[6]

Output vector of the capsules is $s_j = \sum_{i=1}^{m} c_{ij}\hat{\mathbf{u}}_{j|i}$

The squashing function will apply to the output vector to get the activation of $\mathbf{v}_j$

$$\mathbf{v}_j = \mathbf{squash}(\mathbf{s}_j)$$

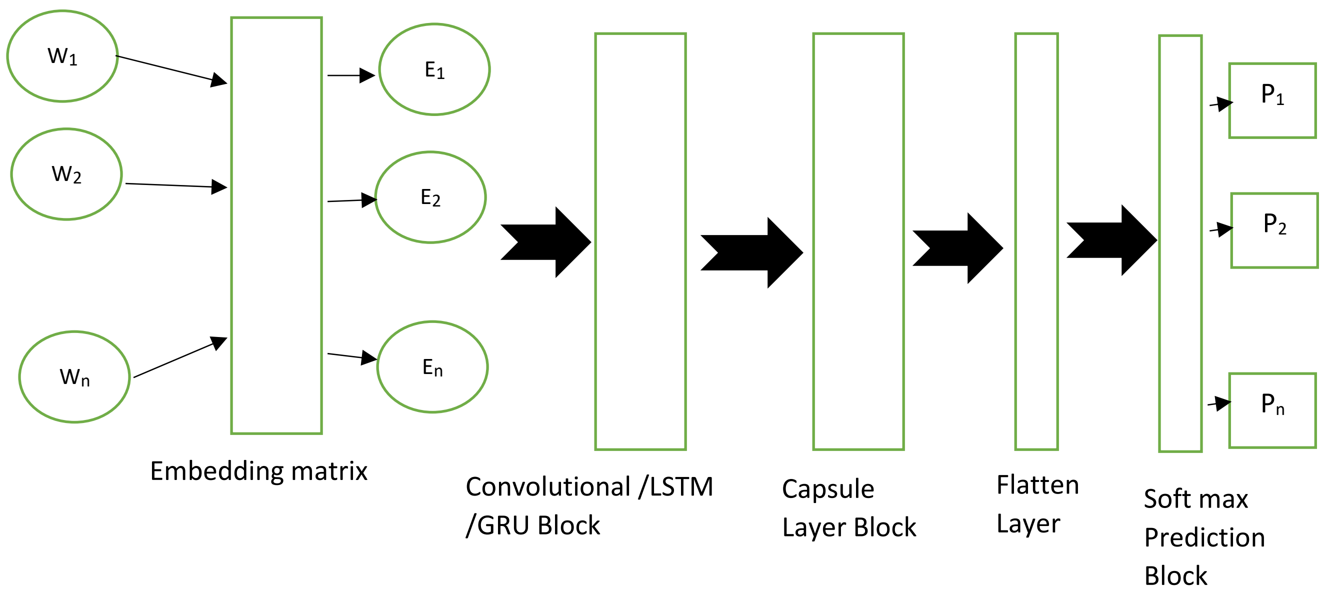

Figure 3.5 Capsule Network Architecture

In capsule network using a dynamic routing algorithm. The activation vectors of layer **l+1** send feedback signals to the capsules at layer **l**. If the prediction vector of capsule **i** (of layer **l**) for a capsule j (of layer l+1) is in agreement with the activation vector of capsule j, their dot product should be high. Hence the "weight" of the prediction vector u***^j|i*** is increased in the output vector of j . In other words, those prediction vectors that helped the activation vector have a lot more weight in the output vector.[6]

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

$$\sum_k c_{ik} = 1$$

Chapter 4: Design

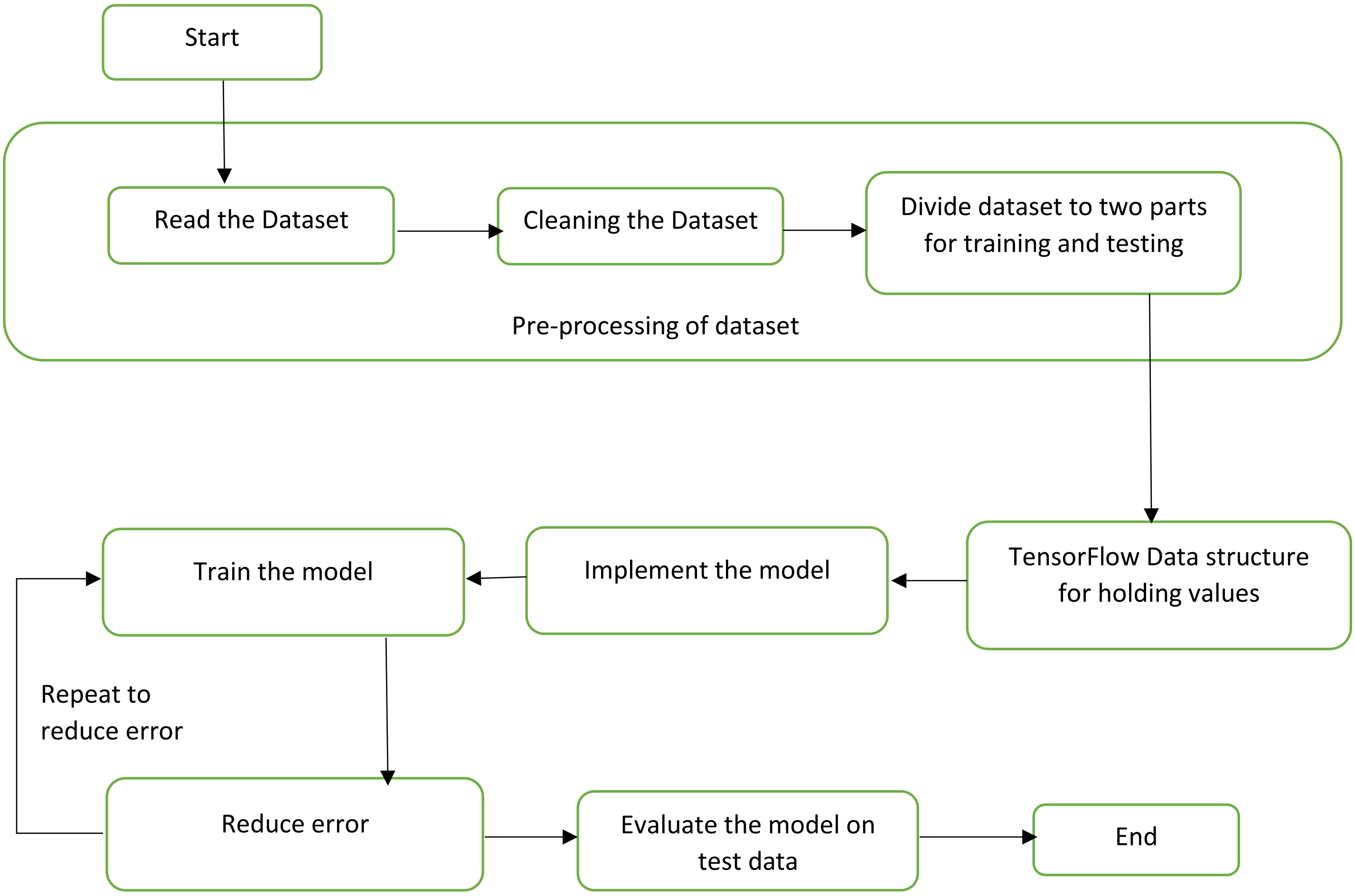

Figure 4.1 Twitter corpus sentiment analysis process flow chart

Figure 8 shows the general procedure that follows in the deep learning models for using the Twitter corpus dataset to analysis the sentiment throughout the research implementation. As first step dataset will be preprocessed and remove the “#” signs and “rt” word that use for mentioning the ReTweets. Then dataset has split into two parts as test and training data. In here we use 67 to 33 ratios between training and test dataset split. Training dataset feed into the implemented model and after the training, it will evaluate the trained model by using test dataset. This gives the accuracy and loss values of the model.

For comparison of the capsule network model, two other models are implementing and feed the same dataset. The results will be compared according to the capsule network model. Other two models are Convolutional neural network model and LSTM network model. These two models get the same parameter as the capsule network model.

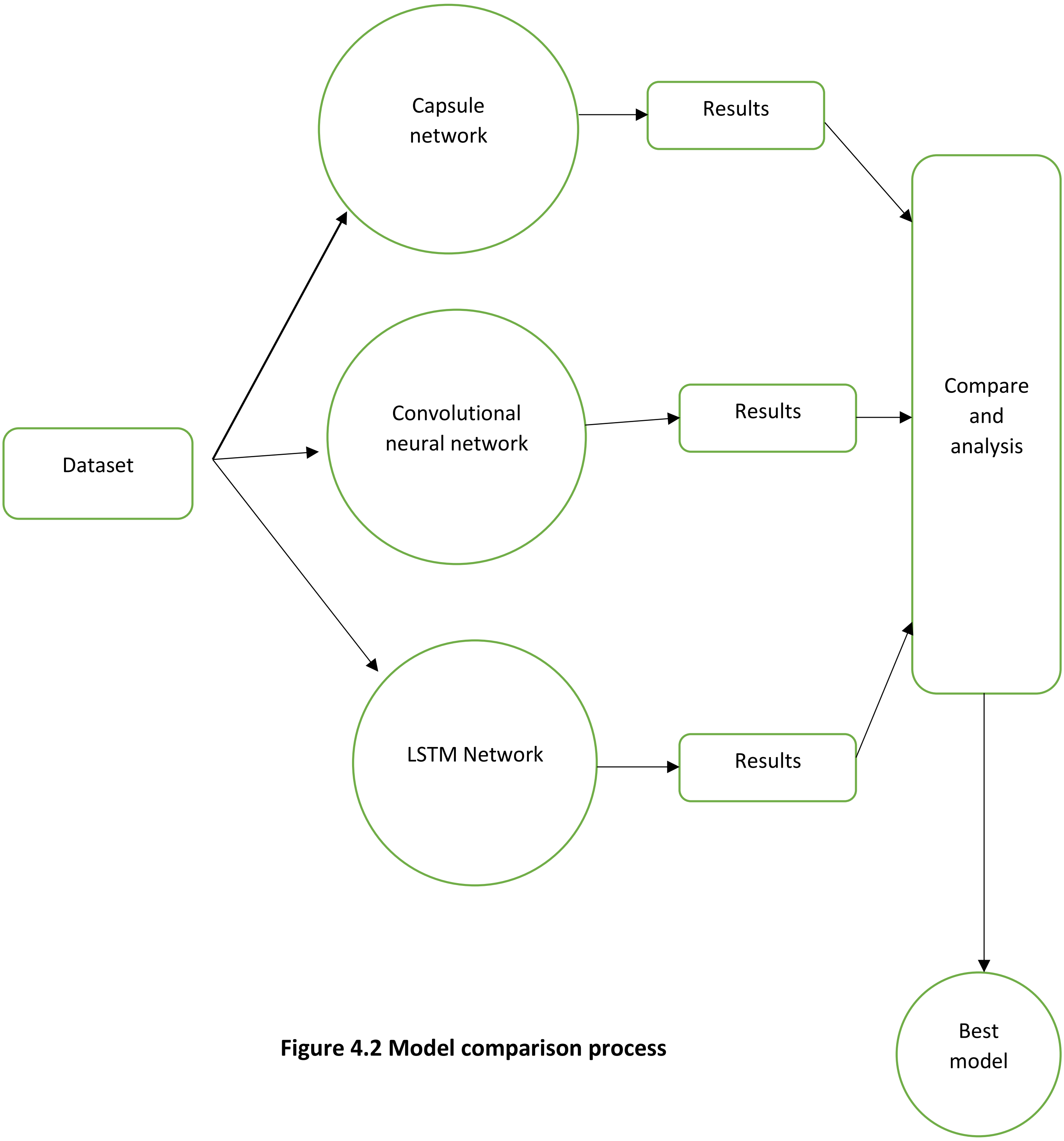

Figure 4.2 Model comparison process

Chapter 5: Implementation

In capsule network, the first layer is the input layer. It takes the preprocessed tensor data into the network. The shape of the input layer is 10. The input layer is connected into the embedding layer for word embedding. This layer helps to reduce the overfitting of the network. Then data will feed into spatial dropout layer. This spatial layer is one dimensional. Then it connected to the bidirectional layer. In bidirectional layer, Gated recurrent unit (GRU) mechanism is used. Between bidirectional and flatten layer comes the capsule layer. Five capsules have used in this model. For routing between capsules three routings have used. For capsule implementation, I have used pre-implemented capsule on Keras by 苏剑林 that can be found on GitHub.[15] Then this layer connected with flattens layer that is a scaler layer. This layer fully connected with the dense layer.

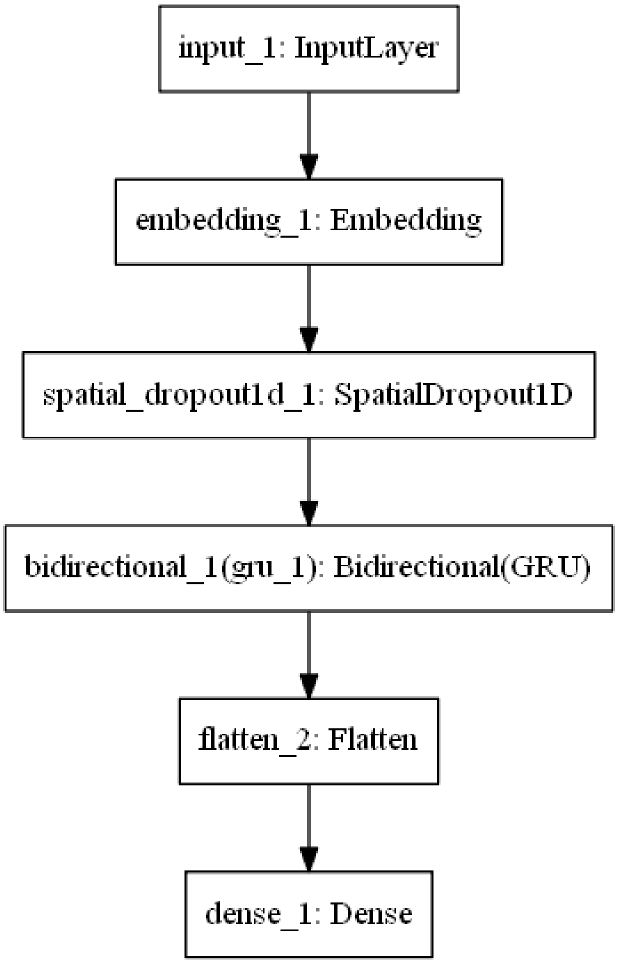

Figure 5.1 Capsule network implementation

Below shows descriptive view of the implemented capsule model.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 10)	0
embedding_1 (Embedding)	(None, 10, 256)	10240000
spatial_dropout1d_1 (Spatial	(None, 10, 256)	0
bidirectional_1 (Bidirection	(None, 10, 512)	787968
flatten_2 (Flatten)	(None, 5120)	0
dense_1 (Dense)	(None, 2)	10242

Total params: 11,038,210

Trainable params: 11,038,210

Non-trainable params: 0

In convolution neural network dataset feed into the input layer. Input layer connected to the embedding layer. Embedding layer connected to the one-dimensional spatial dropout layer. Dropout layer connected to the one-dimensional convolutional layer. Convolutional layer connected to max pooling layer and flatten layer and the fully connected dense layer is connected in the network.

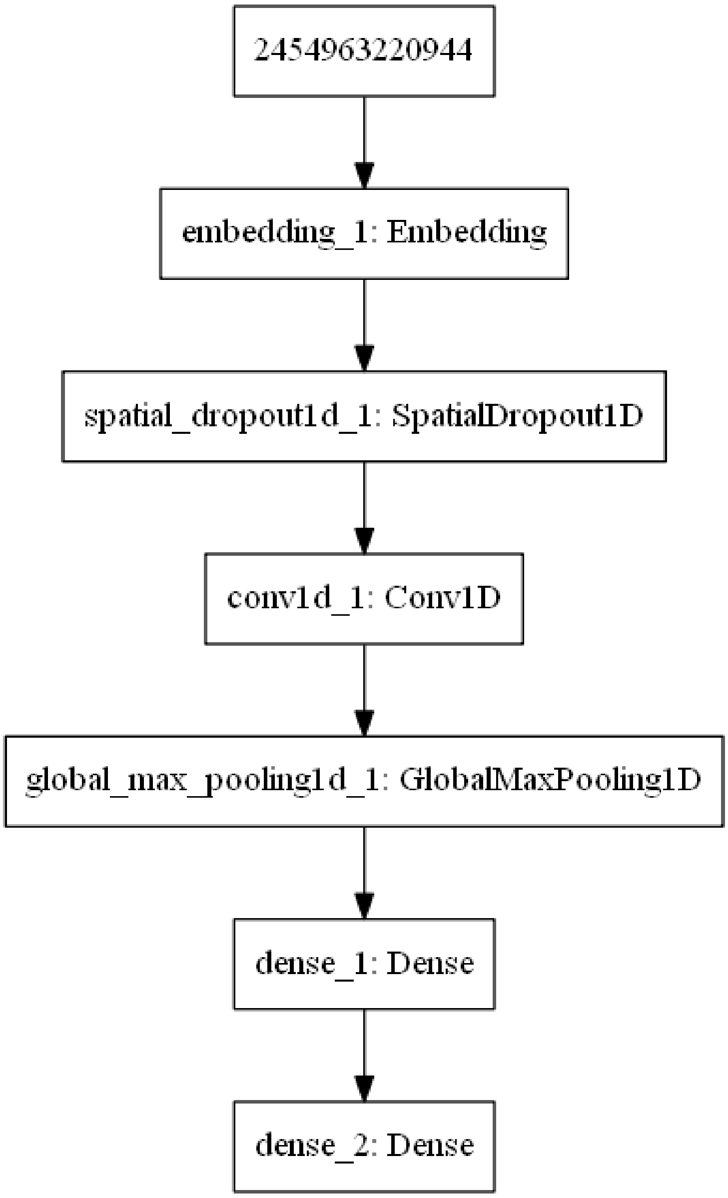

Figure 5.2 Convolutional neural network implementation

Below shows the descriptive view of the convolutional neural network.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 10, 256)	10240000
spatial_dropout1d_1 (Spatial	(None, 10, 256)	0
conv1d_1 (Conv1D)	(None, 6, 128)	163968
global_max_pooling1d_1 (Glob	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290
dense_2 (Dense)	(None, 2)	22

Total params: 10,405,280

Trainable params: 10,405,280

Non-trainable params: 0

Long short-term memory (LSTM) recurrent neural network, dataset feed into the input layer. Input layer connected to the embedding layer. Embedding layer connected to the one-dimensional spatial dropout layer. Then connected to the LSTM layer. In the LSTM layer recurrent dropout is 0.28. LSTM layer connected to the Dense layer.

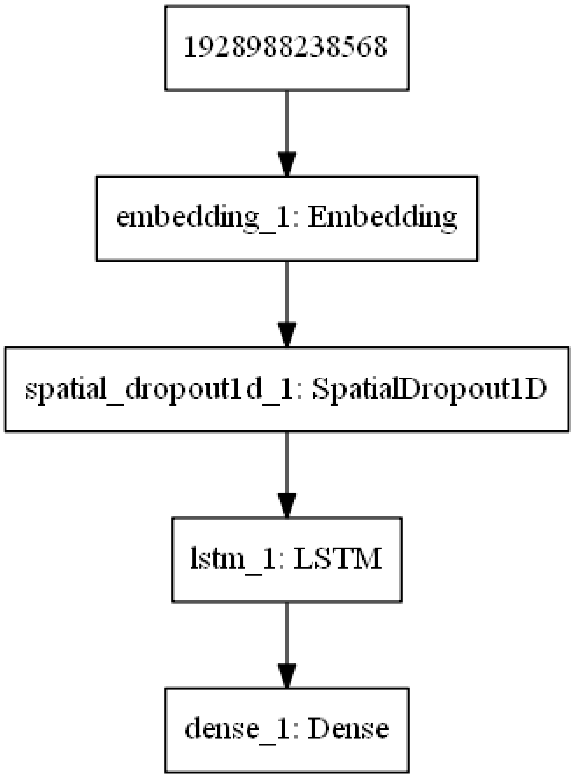

Figure 5.3 LSTM neural network implementation

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 10, 256)	10240000
spatial_dropout1d_1 (Spatial	(None, 10, 256)	0
lstm_1 (LSTM)	(None, 256)	525312
dense_1 (Dense)	(None, 2)	514

Total params: 10,765,826

Trainable params: 10,765,826

Non-trainable params: 0

5.1 Tools and technologies

- **Keras**

  Keras is an open source neural network library written in Python language. It is running on top of the Tensorflow. Using Keras, the capsule network model is implemented. Also, Keras is used for preprocessing the Twitter text before the training process.

- **Tensorflow**

  Tensorflow is an open source software library for dataflow programming and machine learning application. Tensorflow is used as a base of the capsule network implementation. And Tensorboard facility is used to view the network that has implemented on Keras.[16]

- **Scikit-learn**

  Scikit-learn is a free software machine learning library for the Python programming language. Scikit-learn is used for splitting the dataset for training and validation test in capsule network.[17]

- **Matplotlib**

  Matplotlib is a Python 2D plotting library. Using matplotlib result is visualized as graphs and figures for analysis.[18]

- **Pandas**

  Pandas is an open source library easy-to-use data structure and data analysis tool for Python programming language. Pandas is using to read the dataset file and convert data as a data structure that can feed to the capsule network for training and validation.[19]

- A machine that used for model training has processor Intel® Core i5-8250U CPU @1.6GHz 1.8Ghz. As a memory, 8GB RAM has used.

Chapter 6: Evaluation and Results

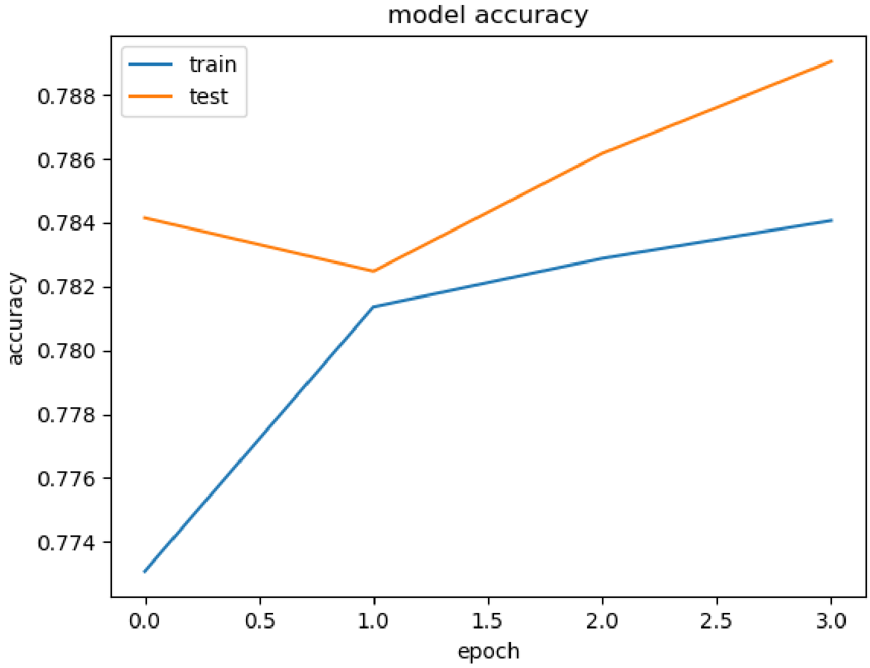

Figure 6.1 Capsule network model accuracy

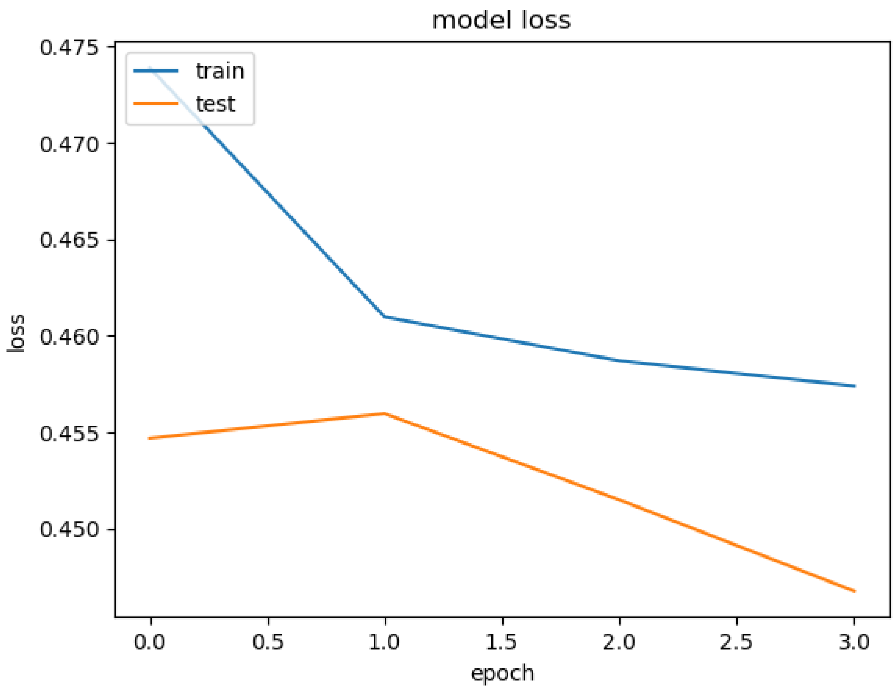

Figure 6.2 Capsule network model loss

```
=================================================================

Total params: 11,038,210

Trainable params: 11,038,210

Non-trainable params: 0

_________________________________________________________________

Train on 1072000 samples, validate on 528000 samples

Epoch 1/4

1072000/1072000 [==============================] - 6182s 6ms/step - loss: 0.4739 - acc: 0.7731 - val_loss: 0.4547 - val_acc: 0.7842

Epoch 2/4

1072000/1072000 [==============================] - 6140s 6ms/step - loss: 0.4610 - acc: 0.7814 - val_loss: 0.4560 - val_acc: 0.7825

Epoch 3/4

1072000/1072000 [==============================] - 6128s 6ms/step - loss: 0.4587 - acc: 0.7829 - val_loss: 0.4515 - val_acc: 0.7862

Epoch 4/4

1072000/1072000 [==============================] - 6120s 6ms/step - loss: 0.4574 - acc: 0.7841 - val_loss: 0.4468 - val_acc: 0.7891
```

The capsule network model has trained on Twitter corpus dataset and it has reached 78% validation accuracy in 4 epochs. It has reduced the validation loss to 44% in training. It took nearly 1.7 hours for one epoch and roughly 7 hours for training the model on GPU.

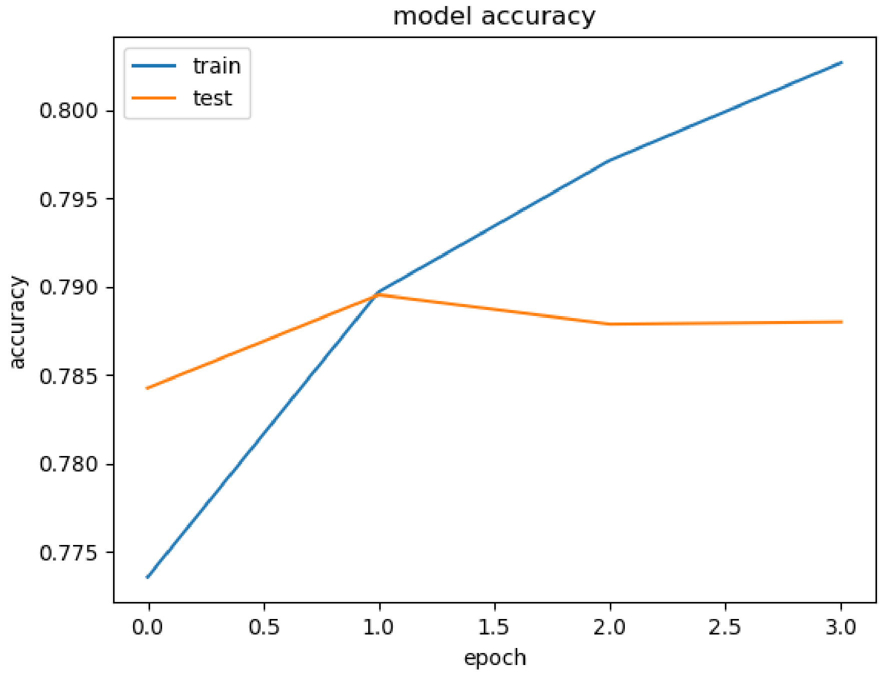

Figure 6.3 CNN model accuracy

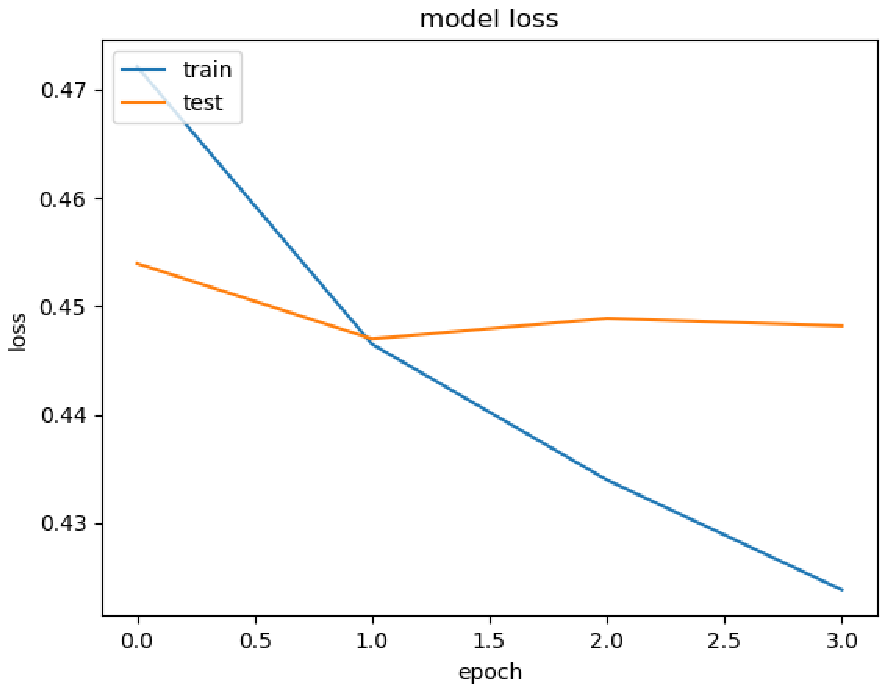

Figure 6.4 CNN model loss

```
=================================================================
Total params: 10,405,280
Trainable params: 10,405,280
Non-trainable params: 0
_________________________________________________________________
Train on 1072000 samples, validate on 528000 samples
Epoch 1/4
1072000/1072000 [==============================] - 4765s 4ms/step - loss: 0.4721 - acc: 0.7736 - val_loss: 0.4539 - val_acc: 0.7843
Epoch 2/4
1072000/1072000 [==============================] - 4765s 4ms/step - loss: 0.4465 - acc: 0.7897 - val_loss: 0.4470 - val_acc: 0.7895
Epoch 3/4
1072000/1072000 [==============================] - 4762s 4ms/step - loss: 0.4339 - acc: 0.7971 - val_loss: 0.4489 - val_acc: 0.7879
Epoch 4/4
1072000/1072000 [==============================] - 4763s 4ms/step - loss: 0.4238 - acc: 0.8026 - val_loss: 0.4482 - val_acc: 0.7880
```

The convolutional neural network model has trained on Twitter corpus dataset in 4 epochs. The model has overfitted the dataset in 2 epochs. It took 1.32 hours to train one epoch and roughly 5 hours for 4 epochs.

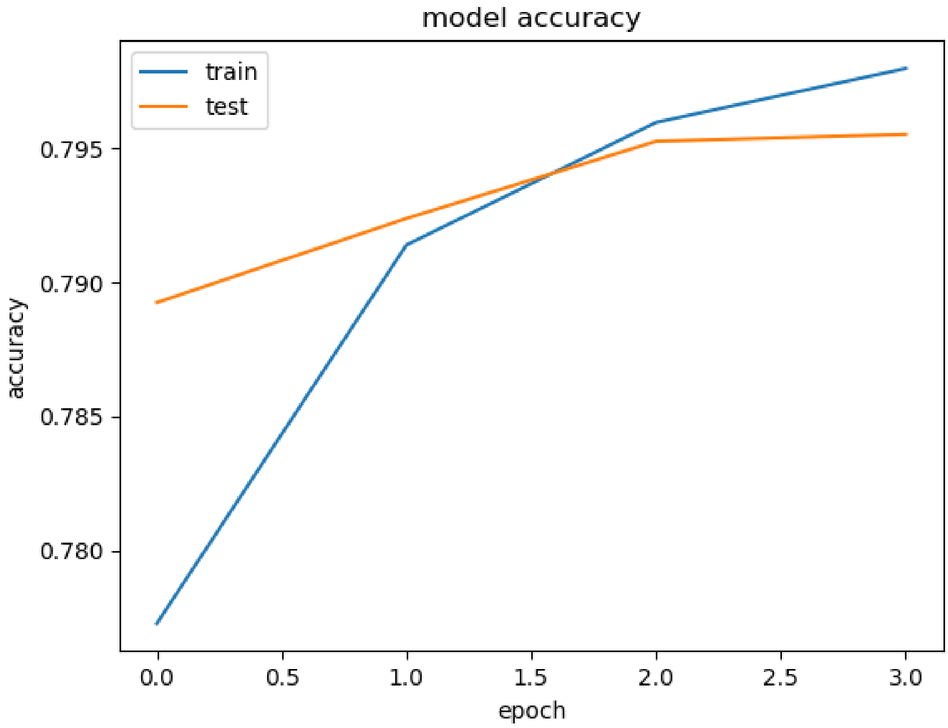

Figure 6.5 LSTM model accuracy

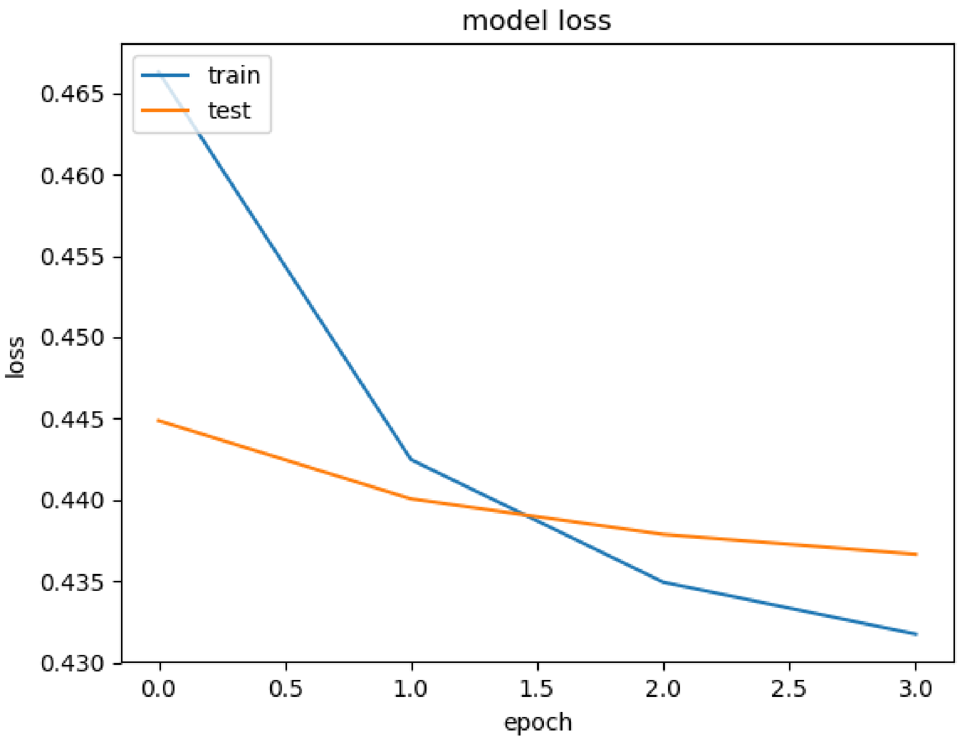

Figure 6.7 LSTM model loss

===

Total params: 10,765,826

Trainable params: 10,765,826

Non-trainable params: 0

____

Train on 1072000 samples, validate on 528000 samples

Epoch 1/4

1072000/1072000 [==============================] - 5794s 5ms/step - loss: 0.4663 - acc: 0.7773 - val_loss: 0.4449 - val_acc: 0.7892

Epoch 2/4

1072000/1072000 [==============================] - 5730s 5ms/step - loss: 0.4425 - acc: 0.7914 - val_loss: 0.4401 - val_acc: 0.7924

Epoch 3/4

1072000/1072000 [==============================] - 5728s 5ms/step - loss: 0.4349 - acc: 0.7960 - val_loss: 0.4379 - val_acc: 0.7953

Epoch 4/4

1072000/1072000 [==============================] - 5720s 5ms/step - loss: 0.4317 - acc: 0.7980 - val_loss: 0.4367 - val_acc: 0.7955

Long short-term memory (LSTM) model has trained on Twitter corpus dataset in 4 epochs. The model has overfitted the dataset in 2 epochs. It took 1.6 hours to train one epoch and roughly 6 hours for 4 epochs.

Epoch	Capsule		CNN		LSTM	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
1	0.4739	0.7731	0.4721	0.7736	0.4663	0.7773
2	0.461	0.7814	0.4465	0.7897	0.4425	0.7914
3	0.4587	0.7829	0.4339	0.7971	0.4349	0.796
4	0.4574	0.7841	0.4238	0.8026	0.4317	0.798

Table 6.1 Training phase results

Epoch	Capsule		CNN		LSTM	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
1	0.4547	0.7842	0.4539	0.7843	0.4449	0.7892
2	0.456	0.7825	0.447	0.7895	0.4401	0.7924
3	0.4515	0.7862	0.4489	0.7879	0.4379	0.7953
4	0.4468	0.7891	0.4482	0.788	0.4367	0.7955

Table 6.2 Validation phase results

Even the convolutional neural network model has gained more validation accuracy over capsule network model, it has overfitted and capsule network model is not. So capsule network model has shown more and good performance on sentiment analysis than the two models over the given Twitter corpus.

Chapter 7: Conclusion and Future Work

The capsule network model has shown more stability and efficiency than the other two models for Twitter corpus dataset. Other two models have overfitted quickly on the dataset, but the capsule model has not overfitted on the dataset.

For future work, can change the dataset and see how this model is behaving for different datasets. In capsule network model we can change the parameters like number of capsules, number of routing, and other hyperparameters and see how the model is behaving and output. Also, can see how the performance of the model on GPU enabled the system. And, we can increase the number of epochs and see how the model behave.

References

[1] P. Basile, V. Basile, M. Nissim, N. Novielli, and V. Patti, “Sentiment Analysis of Microblogging Data,” in *Encyclopedia of Social Network Analysis and Mining*, R. Alhajj and J. Rokne, Eds. New York, NY: Springer New York, 2017, pp. 1–17.

[2] B. Liu, “Sentiment Analysis and Opinion Mining,” p. 168.

[3] V. N. Patodkar and S. I.R, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining,” *IJARCCE*, vol. 5, no. 12, pp. 320–322, Dec. 2016.

[4] A. Severyn and A. Moschitti, “Twitter Sentiment Analysis with Deep Convolutional Neural Networks,” 2015, pp. 959–962.

[5] M. Ghiassi, J. Skinner, and D. Zimbra, “Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network,” *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6266–6282, Nov. 2013.

[6] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic Routing Between Capsules,” *ArXiv171009829 Cs*, Oct. 2017.

[7] “Capsule neural network,” *Wikipedia*. 12-Jul-2018.

[8] E. Xi, S. Bing, and Y. Jin, “Capsule Network Performance on Complex Data,” *ArXiv171203480 Cs Stat*, Dec. 2017.

[9] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, “Sentiment Analysis by Capsules,” 2018, pp. 1165–1174.

[10] S. Joshi and D. Deshpande, “Twitter Sentiment Analysis System,” 2018, vol. Volume 180.

[11] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, “Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification,” 2014, pp. 1555–1565.

[12] M. M. Fouad, T. F. Gharib, and A. S. Mashat, “Efficient Twitter Sentiment Analysis System with Feature Selection and Classifier Ensemble,” in *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, vol. 723, A. E. Hassanien, M. F. Tolba, M. Elhoseny, and M. Mostafa, Eds. Cham: Springer International Publishing, 2018, pp. 516–527.

[13] L. Barbosa and J. Feng, “Robust Sentiment Detection on Twitter from Biased and Noisy Data,” p. 9.

[14] “Sentiment140 dataset with 1.6 million tweets.” [Online]. Available: https://kaggle.com/kazanova/sentiment140. [Accessed: 04-Mar-2019].

[15] 苏剑林(Jianlin Su), *A Capsule Implement with Pure Keras. Contribute to bojone/Capsule development by creating an account on GitHub*. 2019.

[16] “TensorFlow,” *TensorFlow*. [Online]. Available: https://www.tensorflow.org/. [Accessed: 05-Mar-2019].

[17] “scikit-learn: machine learning in Python — scikit-learn 0.20.3 documentation.” [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 05-Mar-2019].

[18] “Matplotlib: Python plotting — Matplotlib 3.0.3 documentation.” [Online]. Available: https://matplotlib.org/. [Accessed: 05-Mar-2019].

[19] “Python Data Analysis Library — pandas: Python Data Analysis Library.” [Online]. Available: https://pandas.pydata.org/. [Accessed: 05-Mar-2019].