DEEP NEURAL NETWORK APPROACH FOR SENTIMENT ANALYSIS OF TWEETS RELATED TO SPORTS CONCUSSION

by

Alex Dela Cruz
Bachelor of Science in Computer Science,
Ryerson University, Canada, 2014

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Science
in the program of
Computer Science

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

DEEP NEURAL NETWORK APPROACH FOR SENTIMENT ANALYSIS OF TWEETS RELATED TO SPORTS CONCUSSION

Alex Dela Cruz

Master of Science in Computer Science
Ryerson University

2018

Abstract

Concussion and Traumatic brain injuries (TBI) are serious injuries that impair the normal functionality of a person's brain. Symptoms can include confusion, disorientation, loss of consciousness, memory lost, and in more sever situations fatality. It is reported that 39% of children (ages 10-18 years old) who visit the hospital due to a sports-related head injury were diagnosed with concussion and 24% with the possible concussion [1]. In order to bring awareness about the seriousness of the TBI to the attention of the policy makers, a neural network based sentiment analysis ensemble system that automates the process of gathering the opinion of the general public is designed. A preprocessing pipeline is proposed that embeds various word-level features into a single concatenated vector. Input vectors are processed by varying Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) Networks. The proposed ensemble system achieves an evaluation score of 62.71% based on its precision and recall, and compares well with other state-of-the art systems.

Acknowledgments

I would like to thank my supervisor, Dr. Alireza Sadeghian, for his continuous support and guidance throughout my academic pursuit of higher learning. The experience and knowledge I have gathered from him have transcended beyond the domain of research and academia, but rather life lessons that have contributed to the individual I am today. I would also like to thank him for the financial support he has provided through the Big Data Research, Analytics, and Information Network (BRAIN) Alliance established by the Ontario Research Fund - Research Excellence Program (ORF-RE).

In addition, I would like to acknowledge the following financial awards, which allowed me to pursue this master's degree: Ryerson Graduate Fellowship and the Queen Elizabeth II Graduate Scholarship in Science and Technology.

I would also like to thank the Computer Science staff members for all their assistance within the past two years. The administrative members: Norman Pinder, Lori Fortune, Alison Gaul, Alex Zheltov, and for their prompt clerical assistance that allowed me to focus on my research. The technical staff: William Zereneh, Ivan Rubiales, Yousif Nakkash, and Misagh Aghajani for assisting and maintaining the resources that allowed me to conduct and complete my research.

I would like to acknowledge the collaboration with Dr. Michael Cusimano and his neuroscience research group (our research partner at St. Michaels' Hospital) that was established and facilitated through the Institute for Biomedical Engineering, Science and Technology (iBEST). In particular, I am thankful to Adriana Workewych for providing the TBI dataset and introducing us to the methodology used to create the dataset.

I would also like to thank my fellow graduate students at Computational Intelligence Laboratory (CI2). Specifically, I would like to thank, Kayvan Tirdad, who I have had the pleasure of collaborating with during this time. I would also like to acknowledge all the research assistants from CI2 (Seden Akman, Emre Ergul, Mahshid Farzaneh, Ngan Bui, Farooq Khan, Nicky Dam, Sohrab Soltani, Khashayar Habibi, Cory Austin) who aided in the enhancement of the dataset.

Lastly, I would like to thank my family for their love and support that has allowed me to pursue all my goals I have had thus far. It is only through your reassurance that I have been able to achieve what I have achieved thus far in my life.

Table of Contents

Abstract iii

List of Tables viii

List of Figures ix

List of Appendices x

List of Abbreviations xi

1 Introduction 1

1.1 Traumatic Brain Injury & Concussion 1

1.2 Policy & Regulation 4

1.3 Motivations 6

1.4 Challenges & the Proposed Approach 7

1.5 Objectives 10

1.6 Thesis Outline 12

2 Literature Review 13

2.1 Background 13

2.1.1 Fully Connected Feed Forward Neural Network 14

2.1.2 Convolutional Neural Networks 20

2.1.3 Recurrent Neural Network 21

2.1.4 Residual Neural Network 26

2.1.5 Temporal Convolutional Network 28

2.2 Related Works 30

2.2.1 Lexicon-Based Approaches 30

2.2.2 Neural Network Based Approaches for SA 31

3 Methodology 34

3.1 Dataset 34

3.1.1 Main Dataset 35

3.1.2 External Dataset 37

3.2 Preprocessing Pipeline 40

3.2.1 Block 1: Cleaning 41

3.2.2 Block 2: Normalization 43

3.2.3 Block 3: Vectorization 44
3.3 Deep Neural Networks 48
3.3.1 Hyper Parameters 50
3.4 Transfer Learning 57
3.5 Ensemble 58
3.6 Summary 60
4 Experiments & Results 62
4.1 Metrics 62
4.2 Neural Network Results 67
4.2.1 Analysis 1 67
4.2.2 Analysis 2 71
4.2.3 Analysis 3 75
4.3 Pre-training & Ensemble Results 77
4.4 Summary 81
5 Conclusion 84
5.1 Contribution 85
5.2 Direction of Future Works 86
Appendices 89
References 97

List of Tables

TABLE 3.1: A SUMMARY OF THE DATASETS. FOR EACH DATASET, THE FOLLOWING INFORMATION IS SUMMARIZED: TOTAL NUMBER OF DATA, TOTAL NUMBER LABELLED DATA FOR EACH SENTIMENT (POSITIVE, NEGATIVE, NEUTRAL), AND THE TOTAL DISTRIBUTION OF EACH SENTIMENTS IN PERCENT 35
TABLE 3.2: DESCRIPTION OF SENTIMENT LABELS OF CONCUSSION DATASET 37
TABLE 3.3: ENCODING TAGS 42
TABLE 3.4: LIST INDICATING NEURAL NETWORK MODELS ADDED TO ENSEMBLE 59
TABLE 4.1: ACCURACY RESULTS SEMEVAL-2016 DATASET. RESULTS OF THE EXTERNAL SYSTEMS FROM THE SEMEVAL-2016: TASK A COMPETITION ARE COMPARED WITH THE RESULTS OF THE NEURAL NETWORK MODELS PRESENTED IN THE CURRENT BODY OF WORK. WHILE A TOTAL OF 34 SYSTEMS CONTRIBUTED TO THE COMPETITION, ONLY THE TOP 15 RANK SYSTEMS ARE ILLUSTRATED IN DESCENDING ORDER OF ACCURACY. [48] 70
TABLE 4.2: ACCURACY RESULTS OF SENTI-TARGET. RESULTS FROM OTHER STATE-OF-THE-ART SYSTEMS ILLUSTRATED IN THE WORKS OF [58] ARE COMPARED WITH THE RESULTS OF THE NEURAL NETWORK MODELS PRESENTED IN THE CURRENT BODY OF WORK. 71
TABLE 4.3: RESULTS OF F1-SCORE ON ALL DATASETS: SPORTS RELATED CONCUSSION (SRC), SEMEVAL-2016 (SEMEVAL), KAGGLE WEATHER (KG), ROTTEN TOMATO (RT), SENTI-TARGET, UCI, UMICH650. 72
TABLE 4.4: F1 RESULTS OF MODELS PRE-TRAINED ON SEMEVAL-2016 AND KAGGLE WEATHER (KG) TRAINED ON THE CONCUSSION DATASET. THE ORIGINAL PERFORMANCE OF THE NOT PRE-TRAINED MODEL IS ALSO ILLUSTRATED FOR EASY COMPARISON. IN ADDITION, THE TOP PERFORMING MODELS ARE BOLDED. 80
TABLE 4.5: RESULTS OF THE ENSEMBLE SYSTEM. 81

List of Figures

Figure 2.1: A graph of a Fully Connected Feed Forward Neural Network. Each neuron is fully connected to the neurons at the proceeding layer. Each connecting edge of the graph indicates the input of a neuron from its connecting neuron. 14

Figure 2.2: Illustration of a 1-D Convolution. 21

Figure 2.3: A graph representing the internal structure of a recurrent neural network. The solid connected edges indicate the forward connection between the neurons. The dash line represents a delayed connection, where the connected data is from the preceding time. 22

Figure 2.4: Graph of an LSTM, illustrating the connecting layers. Solid edges indicate current connection while dotted edges indicate a delayed connection $(T\text{-}1)$. Circular symbols indicate point-wise operation or concatenation. Square symbols indicate network with labelled activation, where Σ is a sigmoid function. [33] 25

Figure 2.5: An illustration of GRU, where reset(r) and update(z) gate are sigmoid functions, nodes are matrix operations, h is the hidden state, and Ø is an activation function. 26

Figure 2.6: An illustration of a residual block within a Residual Network (ResNet) as depicted in [36]. 27

Figure 2.7: Graph rendered from [37] illustrating dilated casual convolution layers from of filter size 3 and increase dilation of [1,2,4]. At the second hidden layer, a neurons field of view is 6 (i.e., its scope allows it two see up to 6 input sequence within the past 28

Figure 3.1: Diagram Illustrating the Preprocessing Pipeline blocks. 40

Figure 3.2: An example of the Hashtag Preprocessing 43

Figure 3.3: Illustration of block 3 and Dimensions of Preprocessing Pipeline. 48

Figure 3.4: Diagram illustrating a single 1D CNN. The layer consists of 2 filters of size 1 and 2 respectively. 53

Figure 3.5: Diagram illustrating a 2 layer stack 1D convolution network. The first 1D convolution has a filter size of 3 generated a 1×3 feature map. A second convolution takes the 1×3 feature map and applies a 1D convolution with filter size 2. 1D max Pooling is then applied after the second convolution producing a scaler value. 54

Figure 4.1: Diagram illustrating the metrics of true/false positive and true/false negative for a specific label 'A'. The circular region indicates data points classified as 'A' (Selected Predictions). The non-selected predictions (regain within the box but not in the circle) are data points not predicted as 'A'. Circular data points, represent ground-truth labels of 'A' and the X points represent ground-truth labels of non 'A'. 63

Figure 4.2: Line graph illustrating the F1-score for each model on the 3 proportional (non-, semi-, fully-) concussion dataset. 77

Figure 4.3: Heat map illustrating the correlation of the models' predictions. 78

List of Appendices

APPENDIX A 89
APPENDIX B 90
APPENDIX C 92
APPENDIX D 95

List of Abbreviations

ASAS	*Automated Sentiment Analysis System*
CDCP	*Centers for Disease Control and Prevention*
CNN	*Convolutional Neural Network*
GRU	*Gated Recurrent Unit*
LSTM	*Long short-term memory*
NBA	*National Basketball Association*
NCAA	*National Collegiate Athletic Association*
NER	*Named Entity Recognition*
NLP	*Natural Language Processing*
NN	*Neural Network*
MTBI	*Mild Traumatic Brain Injury*
POS	*Part-of-speech*
ReLu	*Rectified Linear Unit*
ResNet	*Residual Network*
RMSE	*Root Mean Square Error*
SA	*Sentiment Analysis*
SRA	*Sentiment Ranking Analysis*
SRC	*Sports Related Concussion*
TBI	*Traumatic Brain Injury*
TCN	*Temporal Convolutional Network*

1 Introduction

1.1 Traumatic Brain Injury & Concussion

One major cause of death and disability in the United States is Traumatic Brain Injury accounting for roughly 30% of all deaths caused by injuries. The Centers for Disease Control and Prevention (CDCP) defines Traumatic Brain Injury (TBI) as an injury which impairs the normal functionality of the brain. TBI can be caused by an impact to the head such as a blow, jolt or bump [2]. However, not all instances of an impact to the head results in a TBI, causing some difficulty in the identification of TBI. Therefore, there is the need to categorize TBI based on the severity at which the brain was injured. The severity can range from mild to severe. Mild TBI (MTBI) or commonly known as concussion, is a disruption of normal brain functionality for a short period of time. Either term shall further be used interchangeably throughout the remaining of this thesis. Some symptoms of MTBI can include but is not limited to confusion, disorientation, temporary memory lost during injury, and/or brief loss of consciousness [3]. In contrast, severe TBIs are more discernible due to their prolonged period of disruption to the brain. For example, they can include loss of consciousness for more than 30 minutes, loss of memory for more than 24 hours, and/or penetrating brain injury (i.e., injury which results in an object penetrating the skull and piercing to the brain) [3]. While the distinction of MTBI with more severe instances of TBI may be quite evident, it is more difficult to identify and quantify the occurrence of MTBI, due to its brief period in which a person experiences symptoms. While the tasks of identifying and quantifying MTBI are difficult and a portion of MTBI instances are usually unreported, MTBI is still one of the leading neurological disorders [3]. Therefore, due to its common occurrence, its association to TBI

and the general health of a person, concussion is important topic of research and is the focus of this thesis.

While the area of research within concussion and TBI is broad and encompass multiple discipline from medical, engineering, biology, computer science and data science, we review some outlining problems with concussion to narrow the scope of focus in this work. In 2013, it was reported that TBI related emergency visits, hospitalization and deaths within the United States were caused majority by falls and hits to the head by an object. Among those reported, 47% accounted for falls while 15% accounted for hits to the head [2]. An interesting aspect, that can be inferred based on two statistics, is the commonality of both causes occurring within sports and recreational activities. While sports and recreational activities provide a multitude of health benefits, it cannot be denied that there also exists some health risk depending on the physicality of the activity. The risk of falling or sustaining a hit to the head is very evident in highly physical contact sports and recreational activities such as hockey, football, basketball and soccer, but can also be present in other non-contact sports like horse riding, skiing and gymnastics. This correlation between sports and recreational activities with major causes of TBI and concussion shall aid in narrowing our scope of focus to only TBI and concussion instances that relate to sports and recreational activities.

A motivation to narrowing the scope of this thesis to sports and recreational is based on the number of reported children in the United States that visited the emergency department in 2012. It was estimated that 320,000 children, ages 19 and under, visited emergency rooms due to a sport related injury that included a diagnoses of concussion or TBI. This is an alarming statistic since the brain of children are not yet fully matured and such injuries could impair their development [4]. Another concerning statistics is the rate of these sports related visits between 2001 to 2012. It has been reported that the number of such visits has more than doubled for children 19 years and under

during this time-span [2]. Therefore, there is clear evidence of a growing problem with an increasing risk to youth participating in sports and recreational activities. As such, more attention should be placed on mitigating the potential of sustaining head injuries in sports, especially within youth organizations.

However, sports and recreational activities should not simply be discouraged because of their benefit of providing an environment that promotes regular physical activities within youth and adults. Regular physical activities are a big contributing factor that affect people's health in a positive way. The CDCP has indicated that the inclusion of physical activity within a person's regular routine can reduce the risk of medical concerns. It can reduce the risk of cardiovascular disease by helping reduce blood pressure and cholesterol level. It can also reduce the risk of type-2 diabetes, metabolic syndrome, and some types of cancers such as colon, breast, endometrial and lung cancer. Lastly, they not only increase the chances of living longer by reducing the risk of diseases but by also strengthening an individual's bones and muscles [5]. Therefore, while there is medical risk to sports and recreational activities in the form of concussion, they provide a multitude of health benefits due the physical activity they provide. As such, the simple elimination and termination of sports and recreational activities should not be enforced but rather other guidelines and rules should be developed in order to mitigate the dangers of concussion while maintaining their physical activity benefits.

1.2 Policy & Regulation

Due to the medical risk associated with TBI and the increasing trend of concussion related injuries within sports, it is evident that better policies and regulations are required. Proper regulations and policies can help promote the health benefits of sports and recreational activates while minimizing the health risk of concussion and traumatic brain injuries. While the prominence of a public issue can promote new policy and regulation changes, the general public's opinion is also an additional factor that influences the decisions of policymakers [6]. For example, it is often the situation where multiple concerns must be addressed but limited resources or restrictions prevents addressing all concerns at the same time. In such situations, the public view of people helps determine concerns that are of higher priority to the general public and help lead administrators to focus on concerns that are more precedence to the majority population. Another import aspect to the development of regulations and policies are the impacts it will produce once instigated.

Understanding how the public would react to a new policy and regulation prior to its enforcement is important because a new policy and regulation could incur unintended consequence due to this lack of knowledge. For example, the policy introduced by the National Basketball Association (NBA) in 2005 that placed an age restriction for potential athletes. This has resulted in an increase of prospecting professional athletes attending college for a single year. These young athletes compete in the National Collegiate Athletic Association (NCAA) for the sole purpose of waiting a year to gain eligibility. This has results in a backlash of negative opinion and a discrepancy towards the integrity of student-athlete commitment towards academic due to these extremely talented players non-commitment to attend class nor gain a degree [7].

This rational can also be applied in the case of concussion policies and regulations. For example, if a new concussion policy is implemented that is too extreme from the current policy or too difficult for the general public in a region to follow, it can cause a negative opinion of the policy to the general public and reduce its effectiveness. If the public is unwilling or incapable of following the policy, it can create risk of the policy to be ignored. Understanding this beforehand can help in determining the necessary actions required. If the public is not yet capable of following a new policy, pre-emptive regulations could be implemented first to prepare the general public. For example, a new concussion policy that requires all players to be examined by coaches for potential concussion after an injury could be less effective if the understanding of concussion by coaches can be limited or misconceived. This lack of knowledge could also create the opinion that winning is more important than the risk injury. In such case, the policy may be ignored more often. Therefore, prior knowledge of the opinion of the public could aid in the development of policy that could shift the culture within sports. For example, regulations can be introduced that educate coaches on concussion to help bring awareness of its dangers. Once awareness is established, more focus can be placed on prevention protocols that mitigate the dangerous of concussion, thus, shifting the winning mentality from participating athletes to a more health conscious mentality.

Another application in which the general public's opinion can aid in the improvement of policy and regulation for a given sport is through a comparative analysis with another sport that contains a general positive opinion towards the dangers of concussion. Through sentiment analysis (SA) of different sports (i.e., basketball, hockey, football, etc.), one can identify sports that show a positive concern for concussion. Decision makers may further analyze the policies and regulations within those identified sports, and employ similar policies and regulations.

In conclusion, the public's opinion on sports concussion can bring awareness to policy makers of an important issue which affects the general public. In addition, the public's opinion can be monitored from one sporting event to another, to determine if changes to policies and regulations have provided a significant impact or if the public's opinion has remained the same. For example, the public's opinion on concussion can be analysed from FIFA's 2014-word cup to that of FIFA's 2018-world cup and determine how the opinion may have differed.

1.3 Motivations

The gathering/analyzing the opinion of the general public is in itself a non-trivial task. The research and analysis on the opinion of the general public utilizing traditional methods, such as survey sampling interviews, survey questions, and questionnaire distributions, are costly and time consuming procedures. The gathering of public opinion via in-person and telephone survey interviews requires a large number of human resources and often the survey sample list that organizations utilize to contact the public are outdated [8]. In the case of survey questions and questionnaires, that are distributed via the web or mail, the type of questions asked can greatly affect the type of data that can be gathered, therefore understanding the opinion of the general public regarding different topics would require different sets of questionnaires and can be very complex. Another downside to survey questionnaires is the possible introduction of bias due to the type of questions asked or not asked. For example, questionnaires that focus too much on specific characters of a subject, that tends to be positive, may generate skewed results due to the exclusion of questions relating to negative features that may not have been initially known. Another major challenge faced with traditional survey research, is the high and increasing nonresponse rate. Even if we can identify individuals with the knowledge set or background to complete a survey, it is not

necessarily the case that all the individuals are willing to participate in the survey. Traditionally, there is no immediate benefit, such as monetary reward, for an individual to complete a survey. The on-going trend of busier and busier schedules lead individual's to be less inclined to volunteer. This produces less survey results, since a large part of people are unwilling to cooperate [8], [9].

This thesis shall thus focus on addressing the problem of understanding the general public's opinion of sports related concussion (SRC) to aid in the development of better policy and regulations. We have already illustrated the growing public burden that concussion and TBI have towards our society and the commonality of cases exhibited within sports and recreational activity injures. It is evident that there is an increasing trend of hospitalized visits due to SRC injuries. Thus, there is clear motivation to improve concussion and TBI related policies and regulations within sports and recreational activates. The aim of these polices and regulations are to reduce the health related risk presented in concussion and TBI, while maintaining public interest in regular physical activates offered by sports and recreational activates. There is a motivation to push policymakers, in recognizing the importance of better concussion policies and regulations in sports. This can be achieved by providing evidence that the issue is of importance by analysing the general public's current view point on the subject matter. Thus, this thesis is aimed to identifying a solution to understand the public's opinion on SRC and TBI.

1.4 Challenges & the Proposed Approach

The immergence of social media, within the past years, has generated a new global trend that has resulted in a wealth of information. Among social media platform, Twitter has become a very popular micro-blogging service that has provided a convenient and quick outlet for people to voice

their opinions on any topic they desire. The service has become the number one social media platform within the United States with an average of 330 million active users [10]. This service provides the users the ability to post their thoughts and opinion of any topic via tweets, which consist of short messages that can included hashtag, emoticons, images, and/or videos. Due to the high availability and exponential growth of available un-filtered opinion based data, this thesis shall thus utilize such a rich resource of opinion based data. With an abundant number of users and a constant increase of available tweets to understand the public's views, the analysis now becomes a challenging task. It is no longer feasible for traditional manual methods to determine if the given tweet is depicting a favorable or non-favorable opinion of the user. Thus the need for an automated sentiment analysis system (ASAS) becomes an essential tool to determine the public opinion via tweets.

Sentiment Analysis (SA) is the examination of data to determine the view, opinion or attitude towards a topic or event and has become a popular area of research in recent years [11]. We propose a machine learning method, specifically deep neural networks, to automatically label the sentiment of tweets. Machine learning algorithms have grown in popularity in the past years due to their capability of learning generalized solutions to complex problems. Similar to human behavior, machine learning methods learn the correlation between the problem and answer based on a set of examples. Specifically, for supervised learning techniques the relation between the input data and the output labelled data are learned by training the model with a large set of training examples. Due to this capability of pattern recognition via training samples and the given the complexity of the problem and the success that deep neural network techniques have gained in recent years within complex domains, we propose machine learning methods to automatically determine the sentiment of twitter data. The methods investigated in this thesis shall thus focus on deep neural network

techniques. As such, the main focus of this work will primarily be on convolutional neural network (CNN), recurring neural network (RNN), and feed forward neural network (FFNN).

Another challenge in analyzing twitter data, that must be considered, is the variability in the writing style of different twitter authors. Due to a character limitation, twitter users have become creative in delivering their point across. While some users may be more literal, and their style of posting may be more direct and right to the point. In contrast, some users may be more sarcastic providing more of a cryptic style towards their post, masking their true opinion. Due to this variability in writing style, the level of complexity to analyze one style from another can be greatly different.

An ensemble system produces a final consensus by combining predictive outputs from different models. This notion of 'two heads are better than one', stems from the fact that varying information can be combined to produce better results. Thus, to increase performance further from a single neural network, an ensemble approach is proposed in order to combine the different deep neural network models [12]. The main concept of the ensemble system, is combining different predictive outputs from different models.

Traditional sentiment approach focus on establishing a lexicon to represent the sentiment of a word or combination of words to determine the overall polarity of the message. However, this thesis moves towards a different direction due to the added complexity of understanding the sentiment within SRC. Generally, research conducted on SA analysis is focused on two-levels: document-level or target-level. Document-level analysis attempts to understand the overall sentiment of a document. On the other hand, target-level analysis attempts understand the authors opinion on a specific subject within a document. However, the sentiment problem in understanding public opinion on SRC should be categorized as a variation to comparative SA. Comparative SA, aims at

comparing the sentiment of two entities of the same subject [13]. Comparative SA is often applied to product/move review. While one may argue that comparative analysis is being performed on the sentiment of SRC, there is one key difference between the SA in this thesis and the traditional comparative SA. In the SA of SRC, two completely different entities from different subject matter is being compared. The goal of the analysis in this thesis is to determine if one entity's sentiment is more favorable than a completely different entity. Specifically, we are interested in determining if the authors sentiment on the severity and dangers of concussion is more positive than their sentiment on the game of a given sport. We shall further refer to this newly introduced sentiment as sentiment ranking analysis (SRA).

An important aspect, to analyzing text data for the purpose of SA and the utilization of neural network is the encoding used to represent words within a sentence. As such, the approach to this thesis shall also place emphasis on the encoding methods and preprocessing procedure necessary to embed the text into rich vector representations. Thus a method to combine varying linguistic information into a single vector representation is proposed in this body of work. This thesis shall focus on the following linguistic methods and embeddings: word2vec, part of speech recognition, named entity recognition, sentiment lexicon and sentiment polarity. Due to the unique challenge of twitter data being brief short messages, the embedding and preprocessing procedures need to be normalized to deal with the sparsity within the tweets.

1.5 Objectives

To summarize, there are three key objectives to the body of work in this thesis:

1. **Sentiment Pre-processing Pipeline**: The first objective is to construct the universal pre-processing pipeline that combines essential linguistics and sentiment components into word vector representation that can be feed into different neural network architectures.

2. **Automated Sentiment Analysis System for Sports Related Concussion Text:** The second objective is to implement and evaluate the first SRA system and the first automated sentiment analysis for SRC text. While documents may contain multiple subjects, prior SA has primarily been focus on the general overall sentiment of the document or the identification of the sentiment of a single subject within the document. In either case, the comparison of different multiple subjects is not considered. However, the problem presented within this thesis, contains two main entities (concussion and sports) which are being ranked among each other. Thus, this is the first body of work that presents a system which ranks the sentiment of two different entities.

3. **A methodology for adaptive development of a sentiment analysis system:** Lastly, while this thesis focuses on the development of an automated analysis system for concussion related data. The terminology in different domain problems may differ from concussion data which consist of primarily medical and non-medical terms. As such, the methodology should be adaptive to the development of other sentiment analysis, regardless of the domain problem. Therefore, the basic principles of the preprocessing pipeline, pre-trained models and ensemble approach should be easily adaptable to other problem domains such that hyper-tuning would be the only required procedure.

1.6 Thesis Outline

The remainder of this thesis is organized as follows:

- **Chapter 2:** is a literature review providing a summary of different deep neural network architectures, followed by a summary of related works in sentiment analysis.

- **Chapter 3:** provides background information about the methodology of the current work in this thesis. Firstly, it describes the different dataset utilized within the body of work. Secondly, an illustration of the preprocessing pipeline is provided, followed by a description of each deep neural network model. Finally, an explanation of the pre-training approach and the ensemble system is provided.

- **Chapter 4:** provides the evolution metrics used within the study. It is then followed by the presentation and discussion of the results. The first set of results illustrate the performance of 3 different analysis conducted on the varying neural network. The last set of results provide details of the pre-training and ensemble experiments.

- **Chapter 5:** provides a summary of the work conducted in this thesis. It highlights the main contributions this body of work offers to the growing and popular research are of sentiment analysis in social media. It is then followed by possible directions of future works.

2 Literature Review

This chapter is segmented into two main sections. Firstly, Section 2.1 provides an overview of neural networks and their varying architectures. Secondly, Section 2.2 is dedicated to the related works within SA as to depict the current state of active research relating to this thesis.

2.1 Background

Neural network is a broad area of research; therefore, this section is not intended as a compressive review of this field. Instead this section will provide an overview of the technical theories and algorithms relating to the current body of work. In this section, an explanation of different state-of-the-art neural network architectures are provided, to illustrate the benefits and limitations of each neural network.

Deep neural networks are a family of powerful machine learning algorithms that have generated state-of-the-art results in a verity of problems in recent years. As such, there exists an extensive list of literature explaining the learning theories and algorithms of neural networks. So instead of repeating the process and providing a comprehensive overview of deep neural network. This section is intended to provide an overview of the different unique neural network architectures and their contribution to the work of machine learning. A comprehensive overview of neural network can be found in following literature:

- Schmidhuber's review on neural network provides a comprehensive detail of the development of deep neural network in chronical order [14].

- LeCunn, Bengio, and Hinton's overview provides a comprehensive explanation on Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) [15].

- Goodfellow, Bengio and Courville's book on deep learning provides details on Regularization and Optimization for deep learning [16], [17]

- Ding, Qian, and Zhou's review on activation function provides a comprehensive summary of commonly applied activation functions within neural networks [18].

2.1.1 Fully Connected Feed Forward Neural Network

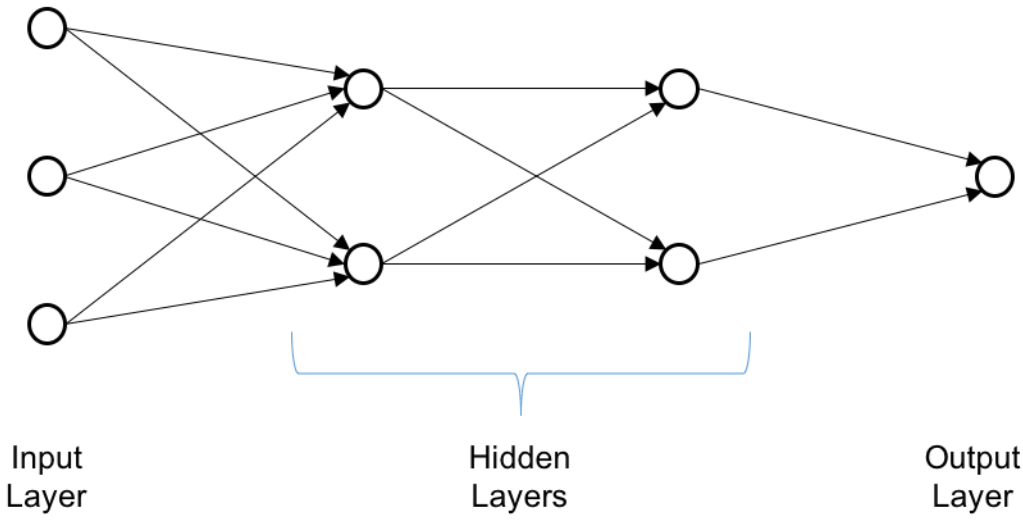


Figure 2.1: A graph of a Fully Connected Feed Forward Neural Network. Each neuron is fully connected to the neurons at the proceeding layer. Each connecting edge of the graph indicates the input of a neuron from its connecting neuron.

Feed Froward Neural Networks (FFNN) are the quintessential building blocks for all sub sequential neural networks. They are fully connected stacked layers consisting of input neurons in the first layer, hidden neurons in subsequent layers and output neurons in the last layer, as

illustrated in Fig. 2.1. The stacking of multiple hidden layers and a nonlinear activation function allows FFNN to estimate a nonlinear function, $\mathrm{F}(x)$. This is achieved by learning a set of weights, called hyper-parameters that represent the weighted contribution of each preceding input to a given neuron. The weighted input can then be summed and passed to a nonlinear activation function to generate the neurons output. This output can be formulated as:

$$a_i^l = \left(\sum_{j \in N^{l-1}} a_j w_{ji} \right) + b_i \tag{2.1}$$

$$z_i^l = \sigma(a_i^l) \tag{2.2}$$

where:

- $a\ and\ z$ are the input and output of a neuron, respectively.
- $i\ and\ l$ are the index of the i^{th} neuron at the l^{th} layer.
- $j \in N^{l-1}$ is the set of neurons from the preceding layer $(l-1)^{th}$.
- w is the connected weight between two neurons.
- b is a bias weight.
- σ is an activation function.

Given (2.2) a generalized form can be written as:

$$z^l = \sigma(X^{l-1} \cdot W^l + B^l) \tag{2.3}$$

where X^{l-1} is a n-dimensional vector, W^l is $(n \times m)$ matrix, and B^l is a m-dimensional vector, such that n is the number of inputs from the preceding layer, and m is the number of neurons at layer l [19].

The weights of a neural network can be learned via gradient descent through the backpropagation algorithm. The algorithm contains the following 3 stages:

1. Forward Pass: In the first stage, observed data is processed in a forward motion, producing a set of predicted outputs.

2. Backwards Pass: In the second stage, the loss function is calculated based on the predicted outputs and the ground-truth. For multi-classification problems, categorical cross-entropy is commonly used as the loss function. The gradient of the loss function in respect to the weights is then calculated to determine how a change in the weights affect the loss function.

3. Update Pass: In the third stage, the weights are updated in the opposite direction of the gradient based on a learning rate [19]–[21].

To prevent the neural network from becoming a lookup table to the training data, also referred as overfitting, popular regularization approach, such as dropout can be applied to neural network models. Dropout mitigates overfitting by randomly dropping or disabling a percentage of neurons causing them not to active. By disabling a new set of neurons after each batch, the network is forced to learn the latent pattern in the training data with only a subset of weights. This causes the network to only adjust a subset of weights during each batch, resulting in an more generalized solution than updating all the weights [22].

A. Categorical Cross-Entropy

The loss function in a neural network is an important function that determines the learning objective of the network. The loss function is the calculated error of the networks prediction with the ground-truth. This is utilized in calculating the gradient of the weight in order to reduce the error of the model's prediction. Cross-entropy is commonly used in classification problems because it measures the error of the model whose output is the probability (a value between 0 to 1) of the observed data belonging to a specific class. That is to say, a model outputting a value of 1 for a specific class, indicates that it is 100% sure the observed data belongs to the given class.

The intuition of categorical cross entropy stems from information theory and gain of information between one probability distribution to another. In neural network, this is the measures of how unique or "surprise" we are to see a specific prediction from the model. For example, a model predicting the probability of *y* as 1 when the true value is 0, is surprising and provides a lot of information within the prediction. In contrast, a model predicting *y* as 0 when the true value is 0, provides no additional information, because the outcome is as expected.

Thus categorical cross entropy can be formally calculated as

$$E = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{K} p_{ij}\log(p'_{ij}) \qquad (2.4)$$

where E is the error of the model, N is the total number of observed data, K is the total number of categorizes, p' is the predicted probability of the model in the range of [0,1], and p is the ground-truth such that

$$p_{i,j} = \begin{cases} 1, & if\ i^{th} data = class\ j \\ 0, & else \end{cases} \tag{2.5}$$

where $p_{i,j}$ will equal one if the ground-truth is category j [19]. Thus, E approaches zero as the model confidently predicts high probabilities for ground-truth classes. Since the $\log(p'_{i,j})$ grows exponentially as $p'_{i,j}$ approaches zero, categorical cross entropy places more weight on confident predictions that are wrong [23].

In the ideal situation, the training dataset would contain an equal representation of each class. However, that is often not the case in real world problems. As such, (2.3) can be extended as followed to alleviate this issue:

$$E = -\frac{1}{N}\sum_i^N \sum_j^K \lambda_j\, p_{ij} \log(p'_{ij}) \tag{2.6}$$

where λ_j is a weighted penalty for class j. Therefore, a weighted penalty higher than 1 would place more weight on incorrect classification of that sentiment. This in turn causes the network to make larger adjustments for misclassifying under sampled classes. Vise-versa, a weighted penalty lower than 1 would place less emphases on incorrectly classifying data of that sentiment.

B. Softmax

Thus, in order to utilize the categorical cross entropy function, the output of the neural network must be the categorical probability distribution of all possible class outcomes. That is to say, the model provides the probability of the data belonging to each class in the range [0, 1]. This can be achieved by pairing the softmax function to categorical cross-entropy [24]. The softmax normalize

the output distribution such that the sum probability of all outputs equals 1 and each output probability is between 0 and 1 [19]. The softmax function can be written as

$$p'_j = \frac{e^{a_j^l}}{\sum_k e^{a_k^l}} \tag{2.7}$$

where:

- p'_j is the predicted probability of class *j*.
- *a* is the input of the neuron as in (2.1).
- *k* is the index of the k^{th} class.
- e is the base of the natural logarithm, otherwise referred to as *e* constant.

C. Limitation

While FFNN are powerful and simple neural networks that perform well on prediction and classification problems of numerical and categorical datasets. There are some limitations that affect performance within highly complex nonlinear problems:

- FFNN does not perform feature extraction, so data engineering should first be performed
- Spatial dependency among input data is not considered. So input data is handled uniformly, such that neighboring data are treated similarly to non-neighboring data.
- Temporal dependency among input data is not considered. For example, the importance of the second word coming after the first word, but before the third word is not maintained. This dependency can be important in natural language processing because two similar text

with different sequential order can be semantically different. For example, “John has a son named Bob” and “Bob has a son named John” would be identical if we just analyze the given words within each sentence. However, we also consider the sequential order the words are presented, then the sentence would be semantically different, since Bob being John’s son is the same as John being Bob’s son.

2.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) have exhibited state-of-the-art perform in both image processing and natural language processing tasks. CNN address both spatial dependency and feature extraction limitation present in FFNN. In the case of sequential problems such as natural language processing, a 1-dimensional convolution is applied instead of the traditional 2-dimensional convolution seen for image processing. In a 1-dimensional convolution, the width of the filter in a convolution is equal to the dimensional size of the vector of a sequence element (i.e., word) and the height is the defined sequence scope of the filter. For example, in Fig. 2.2, input size is $m \times d$ and the 1D filter size is 2, so the window size of the filter is $2 \times d$. Similar to 2D convolution, a pairwise multiplication is applied to the input. Instead of applying convolution across pixels, like in image processing, convolution is applied sequentially across vectors. So in a 1D convolution, filters stride only vertically (row-wise). The principle in applying a 1D convolution instead of a 2D convolution for natural language processing, stems from n-gram representation [25]. By applying convolution with a filter size s, the resulting generated feature

map would be the s-gram representation. A more detailed overview of one dimensional convolution is provided in [26].

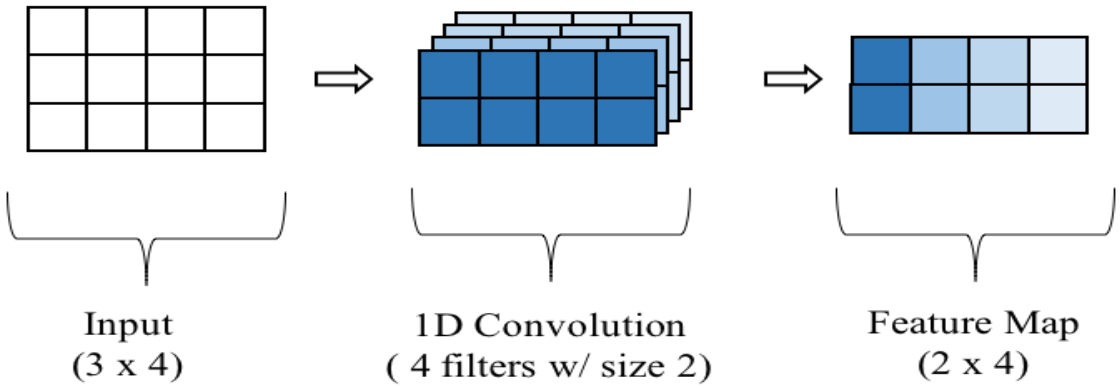


Figure 2.2: Illustration of a 1-D Convolution.

2.1.3 Recurrent Neural Network

A Recurrent Neural Network (RNN) is one that addresses the sequential dependency concern. RNN is similar to FFNN, such that, both contains layers of fully connected neurons. However, RNN contains the added benefit of a recurrent connections, which connects the layers output back to its input. This loop connection allows the network to maintain information from its previous calculation and generate prediction based on prior knowledge, as illustrated in . The neurons in the hidden layer do not just receive the input data but also the predicted output (referred to as hidden state) from the previous time sequence [27], [28].

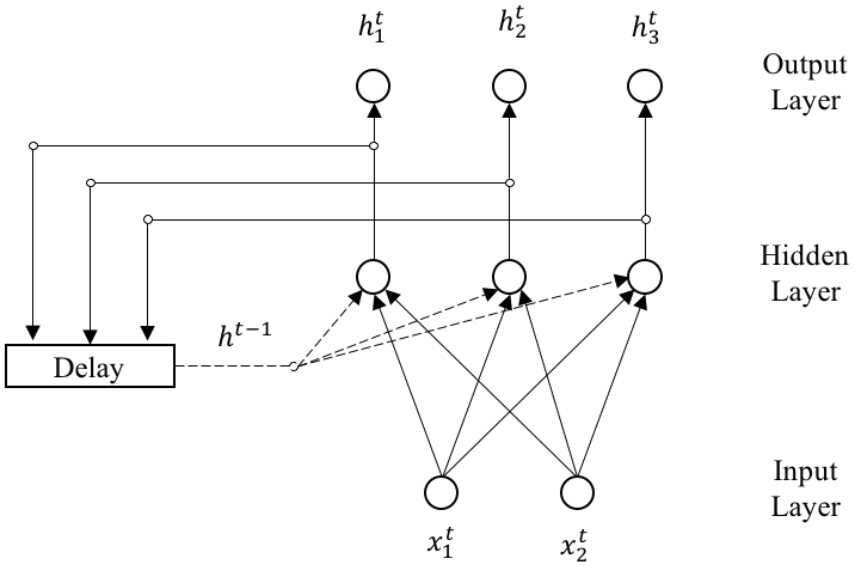


Figure 2.3: A graph representing the internal structure of a recurrent neural network. The solid connected edges indicate the forward connection between the neurons. The dash line represents a delayed connection, where the connected data is from the preceding time.

The unrolling of a recurrent neural network illustrates the network as a feedforward network that spans across the sequence of time [27]. As such, a recurrent neural network can be trained similar to a FFNN by also applying backpropagation but through time [29].

One limitation to a standard RNN, is the limited available data which is presented during training. During training RNN is only capable of retrieving information up to the present future frame. This is commonly addressed by feeding the input in a bidirectional orientation, forward and backwards, known as a bidirectional RNN. More details regarding the implementation of bidirectional RNN is presented in the works of [30].

While RNN addresses the issue with temporal dependency, the network's architecture also makes is susceptive to the vanishing and exploding gradient problem [31], [32]. This is due to the increasing expansion of product of terms within the chain-rule as the sequence of time increases [19]. As a result, RNN has difficulty learning in long sequences. As such, a variance of RNN have been introduced to address the gradient problem and long-term memory limitation.

A. Long Short-Term Memory

One variance to RNN is the Long Short-Term Memory (LSTM) neural network that contains a set of gating mechanics that allows the LSTM to contain a separate memory cell that maintains important information for prediction. This is achieved via three interacting components that controls what information is thrown away, what new information is ignored or retained, and what information is selected for output [27], [33]. [32] provides more comprehensive details of the learning algorithm's forward and backward pass.

As shown in Fig. 2.4, LSTM contains 4 parallel neural network layers that allows LSTM to delegate specific learning task to each individual layer. Different tasks can be achieved by utilizing a combination of different activation functions and merging operations. The following 3 task can be achieved:

- ***What new information to forget:*** is achieved by a neural network layer with a *sigmoid* function. The *sigmoid* function pushes the output to either 0 or 1. This output can be interpreted as the percentage of information to retain from memory, where 1 indicates 100% retention and 0 representing complete discard. Thus information no longer required in memory can then be discarded by applying a pair-

wise multiplication to the memory and forget percentage learned by the network layer.

- ***What new information to ignore/retain***: is learned via two separate network layers. The first layer contains a *tanh* activation to learn possible predictions similar to RNN. The second layer contains a *sigmoid* activation to learn what information to ignore. A pair-wise operation is then performed on both outputs, to produce important information that is retained. This new filter information is then added to the memory via an add operation.

- ***What information to select for prediction***: is delegated by a layer that performs a *sigmoid* function. To mitigate the exploding gradient problem, the memory is first normalized by a pair-wise *tanh* operation on the memory that pushes memory in the range of (-1, 1). A pair-wise operation is then performed on the normalized memory data and selection information, to allow some content from memory to be exposed as the prediction. [27], [33]

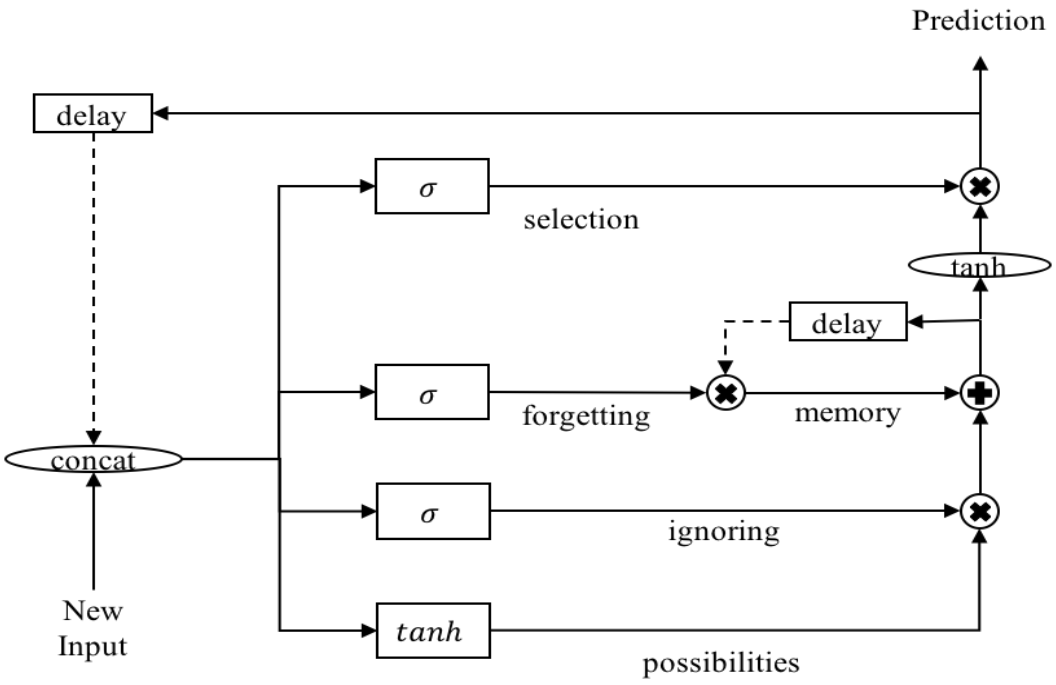


Figure 2.4: Graph of an LSTM, illustrating the connecting layers. Solid edges indicate current connection while dotted edges indicate a delayed connection $(t-1)$*. Circular symbols indicate point-wise operation or concatenation. Square symbols indicate network with labelled activation, where σ is a sigmoid function. [33]*

B. Gated Recurrent Unit

Similar to LSTM, Gated Recurrent Unit (GRU) attempts to learn what information to forget and what information to update. As illustrated in Fig. 2.5, unlike LSTM, GRU does not contain a separate memory cell, but rather the internal hidden state of the GRU itself is maintained and outputted via update and reset gates. Since different hidden units contain separate update and reset gates, the network is capable of capturing a variety of different time dependencies. That is to say, long-term dependency can be achieved by hidden units with infrequent reset activation gates. In contrast, units with frequent reset activation gates will contain short-term dependency [34]. It has

been seen that in some problems, GRU outperforms LSTM both in converging faster and generalization [35].

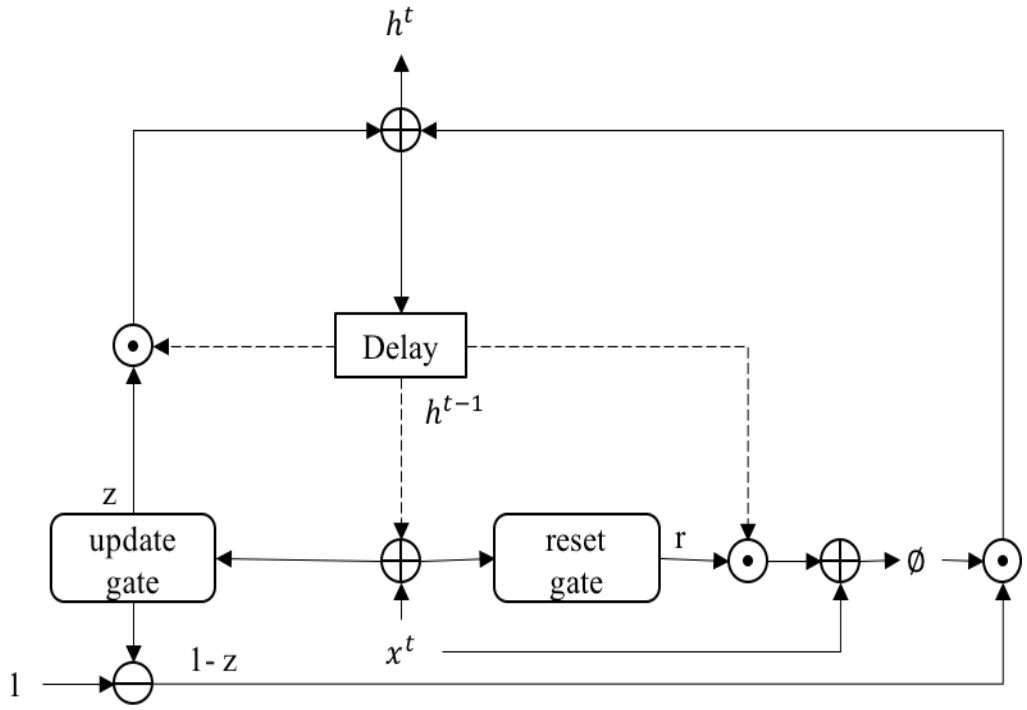


Figure 2.5: An illustration of GRU, where reset(r) and update(z) gate are sigmoid functions, nodes are matrix operations, h is the hidden state, and ∅ is an activation function.

2.1.4 Residual Neural Network

Really deep neural network has been proven to produce the state-of-the-art performance in all areas of machine learning, but as networks get deeper and deeper a degradation problem becomes evident [36]. However, intuition tells us that a network with more layers than a shallower network with similar architecture should perform as well if not better. The assumption is that the same solution that a shallower network is capable of learning is encapsulated within a deeper network. For example, the initial layers of the deeper network can be identical to the shallow network and

the remaining layers simply learn the identity of its prior layer (i.e., the output of the neuron is the same as its input). Therefore, the problem must lie within the training optimization, where layers in the network have difficulty in learning the identity of its input [36].

As such, recent works have introduced residual block layers that change the learning objective of a layer. Instead of traditional layers, where the layers in the network attempt to learn and optimize the original mapping of its input, residual blocks learn and optimize the residual mapping. It can be assumed that in the extreme case such that the input is the optimal solution, it is easier to train the network to push the residuals to zero than to learn the identity mapping by stacked nonlinear layers. As shown in Fig. 2.6, the residual block's architecture is a feedforward network that contains shortcut connections. This shortcut connection allows the network to learn the residual and merge the residual to the identity via element-wise addition.

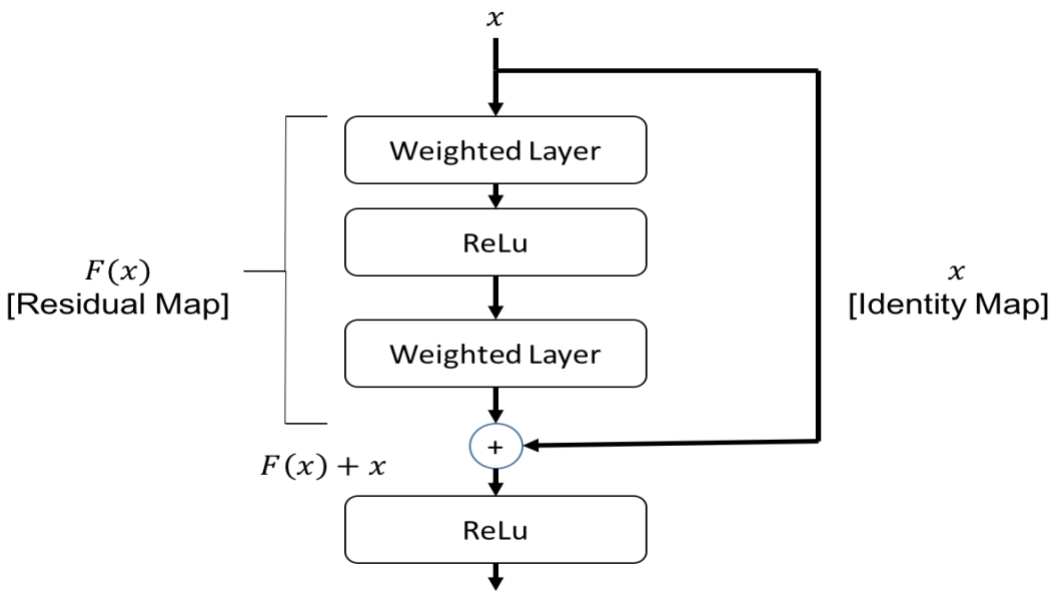


Figure 2.6: An illustration of a residual block within a Residual Network (ResNet) as depicted in [36].

However, the inclusion of shortcut connection introduces an issue with matching dimensions. In the simplest situation, the dimension of the identity and residual map are of the same dimension, forming a solid connection that allows simple element-wise addition to be performed. However, in the case where input and output dimensions differ, a 1×1 convolution can be applied to the identity to project the identity onto the residual [36].

2.1.5 Temporal Convolutional Network

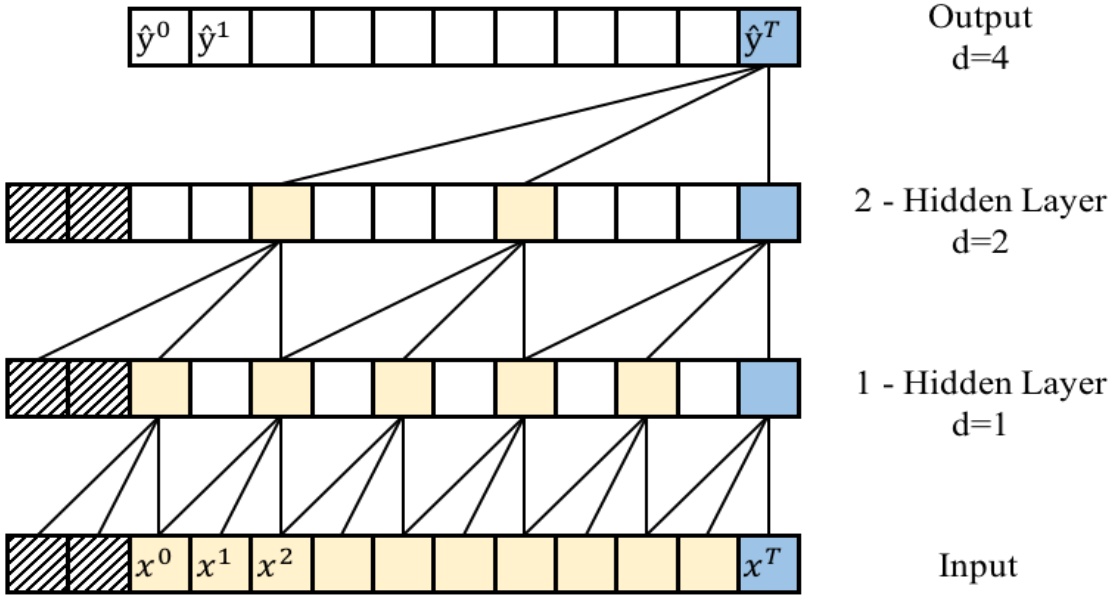


Figure 2.7: Graph rendered from [37] illustrating dilated casual convolution layers from of filter size 3 and increase dilation of [1,2,4]. At the second hidden layer, a neurons field of view is 6 (i.e., its scope allows it two see up to 6 input sequence within the past

While traditional CNN address spatial dependency, there is limitation to temporal dependency within its architecture. As such, CNN has not seen the same dominance in perform in natural language processing task as it has in image processing. Temporal Convolutional Network (TCN)

looks to address the temporal limitation in traditional CNN via attention mechanism that allow convolution to focus on specific region of interest. This is achieved via dilated casual convolutions such that the convolution achieves the following properties:

- larger receptive field than the linear size depth of the network, allowing network to retain longer sequence of history
- "output at time t is convolved only with elements from time t and earlier in the previous layer" [37]

[39] provides a compressive overview of dilated convolution.

It can be seen in Fig. 2.7 that increasing *d* exponentially at subsequent layers, allows the network to retrain longer memory dependency. As such, an increase in network size and/or filter size results in long-term dependency. Vise-versa, short-term dependency can be achieved with smaller filter size and shallower networks. It is however, important to note that longer sequence history via deeper networks can exhibit training optimization problems, as mentioned in section 2.1.4. As such, TCN are implemented with Residual blocks [44]–[46].

The hypothesis in applying TCN in the domain of natural language processing, is the principle that majority of information in a document reside in only certain regions of interest, while the remaining portion only contribute to noise. While readers read a sentence, more attention is generally drawn towards keywords of interest to interpret the semantics that the author is attempting to portray to the reader. TCN attempts to achieve this by learning the attention features from the input via multiple causal dilated convolution layers.

2.2 Related Works

This section will focus on other areas of research within SA in the attempt to illustrate this body of work's position and contribution with that other research that are most similar. This section shall first present related works within traditional SA approaches in preparation for the second portion. The second part then shall focus towards state-of-the-art neural network approaches to SA. While it is noted that traditional DA methods no longer complete with neural network approaches within complex SA task. Related works are presented due to the basic principles and components that are applicable within state-of-the-art solutions. In addition, the inclusion to illustrate works of traditional SA is primarily due to their contributing inspiration to components within the preprocessing pipeline.

2.2.1 Lexicon-Based Approaches

Traditional SA focus on a lexicon-based approach, where a lexicon of n-gram words or characters are built to extract the sentiment polarity of a word. That is to say, a dictionary is created that describes the positive or negative strength for a given entry within the vocabulary. The most simplistic form of generating lexicon is through manual annotation. A method in manually creating a lexicon is presented in the paper [45]. It utilizes MaxDiff type questions to infer polarity intensity of words based on the answered questions, reducing the required number of words to be manually annotated. Through the process of asking which word is the most negative and most positive, one may infer that the remaining words lie between the two annotated words. Through a chain of questions and results, a ranking of the words can be generated. The clear limitation to this process

is the cost problem associated with manual annotation. Therefore, some research have demonstrated automatic approaches to generating lexicons [44]. The automated approaches are all based on measuring the pointwise mutual information of words. So a word can be automatically labelled as positive or negative based on its co-occurrence with other known positive and negative words.

Given the sentiment strength of each individual word, the semantic orientation of a document can then be calculated by comparing the positive terms in the document with negative. The works in [47]–[49] have illustrated the counting of positive terms and negative terms within the document to evaluate its sentiment . The challenging aspect being attempted to be addressed in [50] is the polarity adjustment of terms as they appear in sequence of an influencing term. For example, A. Kennedy et al attempts to address polarity adjustment caused by negations or intensifiers [51]. While P. Chaovalit et al approach focuses on the polarity adjustment of a term based on its semantic application [52]. For instance, a document can contain positive terms, but applied in a factual sense, causing the polarity of the term to decease. More significantly, terms may be applied in an ironic or sarcastic manor, completely reversing and intensifying the polarity of the term.

2.2.2 Neural Network Based Approaches for SA

Neural Networks has seen growing popularity in recent years due to their state-of-the-art performance within a verity of problems. SA is no exception to this rule as witnessed with the recent trend within SemEval competitions. During SemEval 2015 to 2017 message-level sentiment analysis competition , an increasing popularity of deep learning approaches are present among the top ranking teams, with deep learning ranking number one for the last two [47]–[49].

The work of J. Deriu et al presented in SemEval-2016, achieved the highest ranking via a convolutional neural network ensemble system. In their work, they implement, two similar convolutional neural networks containing two layers of convolution and a max pooling layer followed by a fully connected layer and softmax output layer. They differentiated both models via different initialized word embeddings (Google's Word2vec model) and different parameters for convolution and max-pooling (such as filter size) [50].

The system presented by C. Baziotis et al is the only top ranked system that does not utilize an ensemble model. Instead, they implement a single two layer bidirectional lstm network with an attention gate. This allows them to naturally treat the tweet as a sequence of text with lstm while also learning, which region of the tweet to emphasize via the attention gate. Similar to the works of J. Deriu et al, they also utilize a word embedding model (Stanford's GloVe model) to pre-train their embedding layer [53].

M. Cliche et al works was also able to achieve the same top rank as that of C. Baziotis et al. They employed a similar approach to J. Deriu et al by utilizing an ensemble model of 10 CNNs and 10 bidirectional LSTMs. Similar to the other related systems, a Neural network based word embedding model is utilized (Word2vec and Facebook's FastText model). A differentiating factor not present in the other works, is the weighted cost function that places a different weighted penalty for each class. Since the datasets, contains an under representation of negative sentiments, this poses an issue when training the system. As such, this weighing adjusts the cost function, such that under represented classes are penalized more than highly represented classes. This allows the model to pay more attention towards tweets with negative sentiments during training data [54].

It can be noted that ensemble is a very popular approach to sentiment analysis of twitter data. However prior works have primarily utilized neural network models that contained a single vector embedding. Such that the embedding model within their preprocessing stage only captured a single representation of distance based on co-occurrence of words. Therefore, other linguistic features such as part-of-speech, or the semantic orientation are never encoded in the vector representation of a word. Thus the neural network models never see the additional linguistic features. Therefore, this body of work shares similarities to the above related works via the neural network models used and utilization of an ensemble system, but expand of each work by the varying approach applied in this work. This thesis focus on a preprocessing pipeline that utilizes various linguistic features that is applicable to various neural network models. In addition, the ensemble system also utilized a neural network model instead of the traditional 'hard'/'soft' voting algorithms utilized in other works.

3 Methodology

This chapter presents the methods, structures and the processes of implementing the automated system analysis system (ASAS) for sports related concussion (SRC). The material is presented in chronological order to illustrate the sequential building blocks of developing such a system. The first section provides details of the dataset utilized in the implementation of this thesis's main body of works. The following section then presents the components of the system beginning from the preprocessing pipeline, the varying neural network models evaluated for sentiment classification, pre-trained model approach and lastly the ensemble system. For a consolidated list of environment configuration and tools utilized in the implementation of the current body of work, Appendix A may be referenced.

3.1 Dataset

In this thesis two categories of datasets are utilized, the main and the external. The main is the sports related concussion twitter data which relates to this thesis's main problem and objectives. The external is a group of publically available datasets relating to sentiment analysis of other domain problems. These external sets are utilized to evaluate and demonstrate the effectiveness of the proposed neural network designs against other data sets. Table 3.1 illustrates the total number of sample data available for each data set.

Table 3.1: A summary of the Datasets. For each dataset, the following information is summarized: total number of data, total number labelled data for each sentiment (positive, negative, neutral), and the total distribution of each sentiments in percent

Dataset	Total	Positive		Negative		Neutral	
Concussion	*15, 800*	*7, 456*	*47%*	*2, 730*	*17%*	*5, 614*	*36%*
SemEval-2016	*22, 821*	*9, 165*	*40%*	*3, 429*	*15%*	*10, 227*	*45%*
Kaggle Weather	*21, 510*	*8, 772*	*41%*	*6, 931*	*32%*	*5, 807*	*27%*
Rotten Tomato	*56, 912*	*6, 529*	*11%*	*4, 945*	*9%*	*45, 438*	*80%*
Senti-Target	*6, 184*	*1, 536*	*25%*	*1, 538*	*25%*	*3, 110*	*50%*
UCI	*2, 908*	*1, 459*	*50%*	*1, 449*	*50%*	*0*	*0%*
Umich650	*1, 027*	*528*	*52%*	*489*	*48%*	*0*	*0%*

3.1.1 Main Dataset

The main datasets were composed of extending the library of labelled data from the original works of Workewych et. al. [53]. The dataset was provided and made available to us by neuroscience research group of Dr. Michael Cusimano at St. Michaels hospital. A database of twitter data was first establishing by the research group via Twitter search engine. A combination of 18 scientific and 23 colloquial terms were individually searched during June to July 2013. The original works of Workewych et. al. yielded a dataset of 7,483 positives, negative, and neutral tweets. A preliminary investigation and evaluation of the neural network models on the original dataset from St. Michaels hospital was performed. Since the original dataset contained multiple duplicate re-tweets the dataset was first filtered to include only unique tweets, which reduced the dataset to

5,478 tweets with a distribution 2,469 positive, 892 negative, and 2,117 neutral sentiments (45% positive, 16% negative, 39% neutral). However, the initial performance on the original dataset yielded low performance across various neural network models. Since neural network models traditionally performance better with larger sample dataset, effort was place on expanding the original dataset. Additional tweets were gathered using Twitter's Search API [54] for the month of May 2018 that were then manually filtered and labelled by a group of 9 volunteers from the CI2 Lab. The search queries used to retrieve additional tweets contained a combination of scientific (i.e., TBI, concussion, etc…) and colloquial terms (i.e., out cold, clocked out, etc…). The terms were derived from a code book that was initially developed in the works of Workewych et. al. [53]. Each tweet where manually labelled by at least two people, to ensure exclusion of bias that may be introduced by just a single labeler. Indecisive or split labelled results were further labelled by additional labelers until a majority voting of 50% or more was achieved. Firstly, the tweets were labelled based on relevance such that tweets not discussing sports related concussion were discarded. Afterwards, tweets were labelled based on their sentiment ranking between concussion and other sport's related topic. For example, tweets that indicated an awareness of concussion and prioritized its significance over other topics occurring in that sport were labelled positive. A more detailed description of the three levels of sentiment are provided in Table 3.2. To aid in the manual labelling, an application was developed to reduce the burden on labelers by streamlining the process. The application contained a subset of tweets that were presented one by one to labeler to label. For each presented tweet, the application first asks the labeler to label the relevance of the tweet. Relevant tweet is then asked to be labelled based on their sentiment and non-relevant tweets are quickly completed. Afterwards, the next tweet is presented in the same manner. The additional labeled tweets were then combined with the original dataset. The consolidated dataset was then

filtered to only unique tweets by discarding duplicated retweets. At the end, the concussion twitter dataset was expanded to 15,800 tweets.

The distribution of the tweets' sentiment is illustrated in Table 3.1. The distribution indicates an unbalance distribution with the sentiment labels, most notably the distribution between positive and negative sentiments. This distribution can pose a concern during training of a neural network and is discussed in more detail in section 3.3.

Table 3.2: Description of Sentiment labels of Concussion Dataset

Sentiment	**Description**	**Example**
Positive	*The author illustrates the severity of concussion and its superseding importance above all other sport's related topic*	• RT@[user]: Playing on with a concussion isn't big or brave, it's sheer stupidity. (#lions) • @[user] I know. Heard that he has a mild concussion. MLB needs to find a solution before something terrible happens.
Negative	*The author illustrate a degrade, lack of concern or understanding of the dangerous of concussion and/or places more positive sentiments towards other sport's related topic (i.e,. result of the game)*	• I'm so glad Silva got his bell rung • I really enjoyed seeing Anderson Silva get his clock cleaned
Neutral	*The author provides no opinion about any given topic or does not illustrate a stronger positive opinion of one topic to another.*	• Toews is definitely concussed • Cobb leaving hospital, placed on concussion list

3.1.2 External Dataset

A total of six different publicly available datasets are utilized in evaluating each standalone deep neural network model. In addition, two of the datasets containing twitter data were then used to pre-train the deep neural network models within the ensemble model (further discussion in section 3.3). The datasets included in the external are as follows:

- **SemEval-2016**: The dataset contains a set of labelled tweets classifying the general sentiment of the tweet into 3-levels (‘positive’, ‘negative’, or ‘neutral’). The dataset consists of tweets that have been gathered between July–December 2015. Tweets were filtered based on the topics being discussed where only the top 200 most popular topic during that time span were kept. The dataset is made available by the semeval-2016 Task 4: Subtask A competition. They were acquired using SemEval’s download script that utilized Twitter’s Search API [58]. Tweets no longer publically available via Twitter API were discarded. It can be noted from Table 3.1 that the SemEval-2016 dataset bares the most resemblance to the concussion dataset based on their sample distribution (i.e., they contain similar distribution among ‘positive’, ‘negative’, and ‘neutral’ tweets).

- **Kaggle Weather**: The source dataset contains weather related tweets that are classified into five sentiment categories (‘unknown’, ‘not relevant’, ‘positive’, ‘negative’, ‘neutral’) [60]. The tweets contain the user’s opinion of the current, past or future weather. Therefore, a positive sentiment of the tweet would indicate that the user is happy about the weather condition. In order to mold the dataset to conform to the current problem domain. The dataset was filtered to only tweets containing relevant known sentiments by discarding ‘unknown’ and ‘non relevant’ tweets.

- **Rotten Tomato**: The source dataset contains a corpus of movie reviews that are labelled into five fine-grained sentiment labels (‘negative’, ‘somewhat negative’, ‘neutral’, ‘somewhat positive’, and ‘positive’). The dataset contains writing styles that introduce the challenges of negation, sarcasm, and ambiguity. The dataset was then filtered to only 3-levels of sentiment (‘negative’, ‘neutral’, and ‘positive’) in order to resemble the main dataset. The source dataset can be acquired from [61].

- **Senti-Target:** The dataset contains a set of tweets that are labelled into 3 sentiment levels (‘positive’, ‘negative’, or ‘neutral’) for a given target within the tweet [58]. As such, the dataset also contains, the target-topic of the tweet as an additional input. While the dataset is geared towards target-level sentiment analysis and the current sentiment problem of this work is document-level, the additional input is simply excluded such that the system can attempt to predict the sentiment without the additional information.

- **UCI:** The dataset contains review sentences from ‘imdb’, ‘amazon’, and ‘yelp’. The sentences are labelled into 2-levels (‘positive’, or ‘negative’). The dataset was originally created for [59] but is still available via UCI’s machine learning repository [60]. Since the size of the data from each review source is very small, in this body of work, the reviews of all 3 sources are consolidated to produce a single dataset. The varying review domains produces a dataset which contains reviews for movies (imdb), products (amazon) and services (yelp).

- **Umich650:** The dataset contains labelled sentences of (‘positive’, or ‘negative’) that where extracted from social media blogs. The dataset was originally housed at ‘opinmid.com’ but is now available via Kaggle competition hosted by Michigan University [62]. The source dataset contained multiple duplicate sentences, so the dataset was filtered to only unique entries.

Since no publicly available concussion twitter dataset exist, the above six datasets which have already been labelled based on their sentiment are utilized in this thesis. While the subject of each dataset differ from the concussion dataset, the characteristics of the data are similar. In the instance of SemEval-2016, Kaggle weather, Senti-Target, the dataset was also generated from extracted

twitter post. Therefore, the concussion dataset and the three are all twitter data, but with different subjects. As per Umich650, the dataset also comes from social media blogs, so it contains the same informal unstructured characteristics as the twitter data. Lastly, since Rotten Tomatoes and UCI consists of user review, they are also informal in nature. However, they are traditionally longer in length than twitter posts which allows the evaluation of the preprocessing pipeline towards datasets where the document length are much larger.

3.2 Preprocessing Pipeline

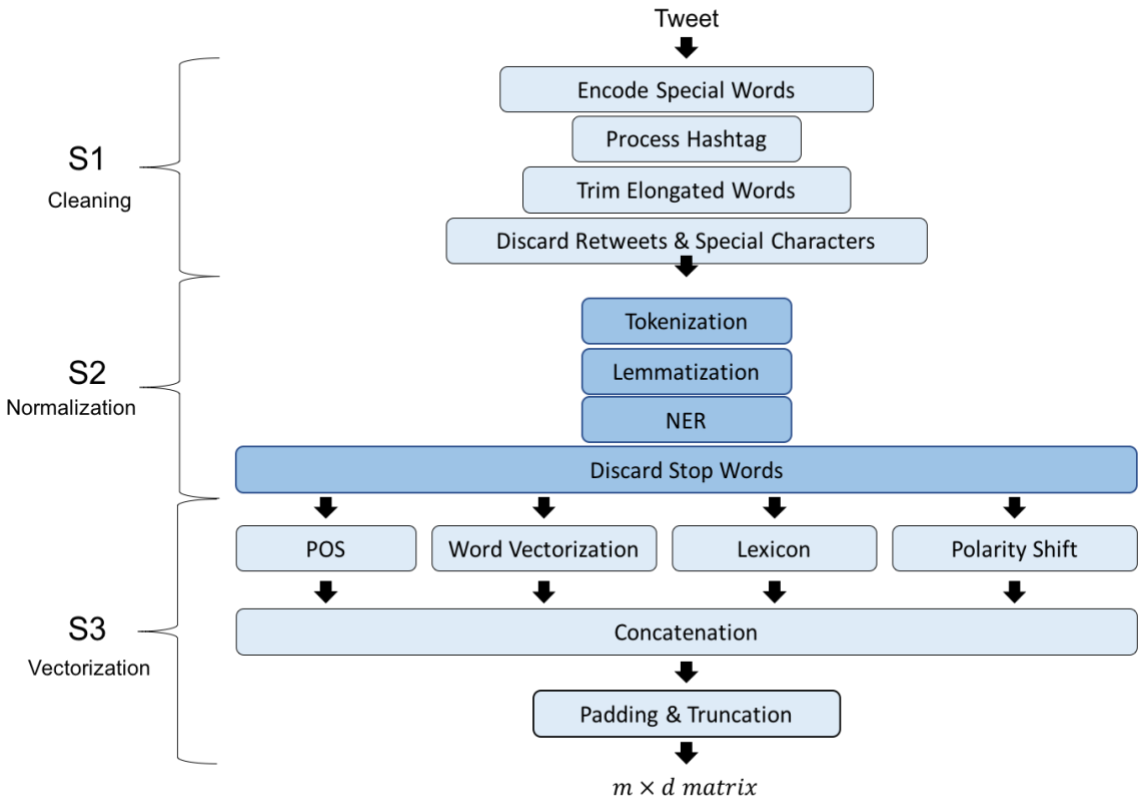


Figure 3.1: Diagram Illustrating the Preprocessing Pipeline blocks.

This section provides an explanation of the methods applied to the tweet during each section of the preprocessing pipeline. The preprocessing pipeline contains three main blocks as illustrated in Fig 3.1 . The first block performs general natural language preprocessing that attempts to mask and clean the tweets. The second block performs a sequence of methods to normalize the tweet. Lastly, the third block in the pipeline performs vectorization of the twitter data to convert the terms into vector embedding representation.

3.2.1 Block 1: Cleaning

Tweets are informal short and quick posts that are often not reviewed prior to posting. There are also different types of tweet posts: communication based (where tweet is addressed to another specific user), general (where author is expressing their opinion to the general public), and retweet (where user is posting about another user post). As such, tweets often contain slight differences due to elements specific to social media (i.e., inclusion of emoticons) and the different types of tweet. These minor differences do not contribute to the semantics of the tweet and only may introduce unnecessary noise. Stage 1 is responsible for cleaning tweets and removing such noise.

First, special words are encoded because the specific differences among these words do not contribute to the semantic orientation of the tweet. For instance, in a communication based tweet, the sentiment is not influenced by who the tweet is being addressed to (i.e., the sentiment remains the same regardless if the tweet is address to user A or user B). As such, the user names are encoded to the single tag '<user>'. A complete list of encoding performed in this stage is provided in Table 3.3.

Afterwards, hashtags within tweets provide significant information about the topic of the tweet. Users will generally add hashtags to tweets, that categorize and emphasize the topics the user

wishes other readers to focus on when view their tweet. As such, an additional preprocessing is performed on hashtag to allow the network to differentiate hashtag words with normal words within the tweet. The preprocessing algorithm for hashtag, performs segmentation and encoding that is best illustrated in the Fig. 3.2. First, the hashtag symbol extracted and encoded with the unique tag '<hashtag>'. Second, the hashtag context is segmented into individual words resulting in an output of the encoded hashtag followed by one or more words.

Table 3.3: Encoding Tags

TYPE	**EXAMPLE**	**ENCODING**
URL	*http://t.co/sUXrGDXhpm*	*<url>*
Username	*@SampleUser*	*<user>*
Emoticon	*:-[) | <3 | 8\(| ;-p+*	*<smile>*
	<3	*<heart>*
	8-(	*<sadface>*
	;-p+	*<lolface>*
	:- |	*<neutralface>*
Numbers	*3.52*	*<number>*

Lastly, block 1 shortens elongated words and removes retweet indicators and special characters. Elongated words are shortened because the decision to use elongated words is more of a personal preference that reflects the author's writing style than contribute to the semantics of the text. For example, 'omggggg' has the same meaning as 'omg'. The length of elongating a word is not standardized and can differ from tweet to tweet that causes unnecessary sparsity in the dataset. For example, the tweets 'I am soooooo happy' and 'I am soo happy' are semantically the same but a model may not see them the same due to the varying length of the elongated word 'so'.

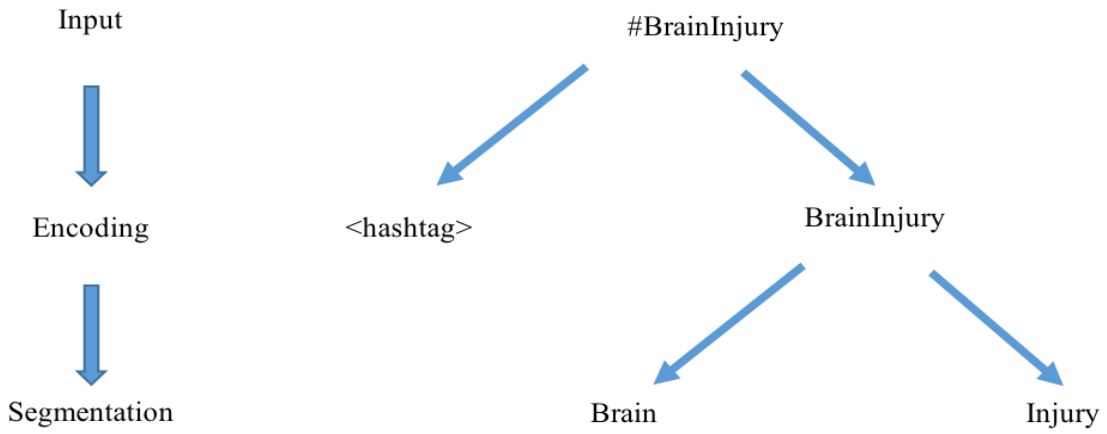


Figure 3.2: An example of the Hashtag Preprocessing

3.2.2 Block 2: Normalization

Multiple languages contain different forms of a word for grammatical reasons [64]. This, however, introduces a sparsity problem within natural language processing. Another problem that lies with the sparsity problem is the distribution between common words with that of non-common words. As such, this block in the preprocessing pipeline aims at mitigating the sparsity and distribution problem by normalizing the data in tweets.

This is achieved by leveraging Stanford's CoreNLP to perform tokenization, lemmatization, and named entity recognition (NER). First, tokenization is used to extract each word within the tweet. After each word is extracted, lemmatization is used to convert each word into its base form. The decision to use lemmatization, instead of stemming was in the aim of ensuring additional noise is not introduced in the process. Therefore, words (such as 'saw') would not be simply cropped in the case to provide inaccurate base form, but rather be converted to their lemma ('see' or 'saw' based on part of speech) [65]. To further normalize the tweets, the name of things, such as: a

person, a location, or an organization etc., are recognized and encoded to a standard term. For instance, the name Bob and John would both be encoded to simply '<person>'. This further reduces unnecessary noise which the neural network model could ignore. The following 7 named entities are identified and classified in this current body of work: name, location, organization, money, percent, data, time [66].

To further normalize the data and address the distribution problem, highly frequent words known as stop words are removed within the tweet. Stop words (such as 'I', 'The', 'A', etc…) appear very frequently and are very useful for grammatical reasons, they however provide little to no contribution to the semantical orientation of the tweet. This is primarily due to their common appearance in different tweets, resulting in no significant information gain in differentiating the sentiments of tweets solely by the appearance of these common terms. As such these words are removed using nltk's stopwords corpus.

3.2.3 Block 3: Vectorization

The final block of the preprocessing pipeline is the section of the pipeline that converts texts into vectors. To achieve the preprocessing pipeline and development methodology objectives, stated in section 1.5, the final block of the pipeline should contain two essential characteristics:

- Interchangeable word embeddings. The pipeline should be capable of seamlessly adding or removing additional word embedding models. Different word embedding models, produce different features that may be more suitable for one problem than another. As

such, the pipeline should be able to allow interchangeability with varying word embeddings.

- Vector structure of tweet should be independent of the receiving neural network model. This is to ensure the same vectored input data can be accepted by varying neural network models, eliminating the need for separate preprocessing for neural network models with different architectures (i.e., FFNN, RNN, or CNN).

First the interchangeable word embedding can be accomplished by performing parallel word embeddings on each word and concatenating each output vector. This produces a single consolidated word vector that contains components that focus on specific features of the word. In this implementation of an ASAS for concussion data, the following embedding is utilized:

- **Word Vectorization:** The motivation to utilize word vectorization, is to acquire a distributed representation of the distance between words. In word vectorization, distance between words are calculated based on the co-occurrence of words within a threshold scope, often a window size of 5. Therefore, a word vectorization embedding attempts to cluster vector terms based on their prior 2 and next 2 neighboring terms. This in turn should generate vector embedding's, such that the vector of semantically similar terms like 'man' and 'women' or 'boy' and 'girl' cluster with equal distance. For example, given the vector of each word, the equation, 'man – woman = boy - girl', should hold true. That is to say, the vector distance between 'man' and 'woman' is equal to that of 'boy' and 'girl'. Through this distance representation, it may be understood that 'man – woman + girl = boy' or more formally as 'girl is the equivalence of boy, as woman is to man'. In this body of work, the pre-trained GloVe model that was trained with 2 billion tweets is utilized to embed each

word within a tweet [41]. The option to utilize the Stanford's GloVe model instead of Google's word2vec model, stems from the nature of the training data that was used to train the model. Since, the word2vec was trained on the Google News corpus, the vocabulary and characters of the data would be more formal than that of the twitter data.

- **Part-of-Speech Embedding:** Since words have different functions based on how they are used, their distinct meanings change based on their function. As such, Parts-of-speech (POS) categorizes these functions into a set of distinct grammatical classes. Since a word's function can change the semantic meaning of a sentence, the inclusion of a POS embedding can increase performance [65]. The inclusion of POS tagging is also important in the Lexicon Embedding phase as we will see in the following point. In the current implementation, Parts of Speech analysis is performed on each word via Stanford's CoreNLP. Each word is annotated with the abbreviated tag used in the Penn Treebank POS English tag set [66]. A list of the part-of-speech tag can be reference in Appendix B. Vectorization is then performed on the abbreviated tags by assigning a unique scaler value between the range (0, 1). The unique value is calculated, by dividing the index of the POS tag to 36 (the total number of POS tags).

- **Lexicon Embedding:** The SentiWordNet 3.0 is directly used as the sentiment lexicon because it not only provides the sentiment polarity of a word but outputs the probability distribution of the word being 'positive', 'negative', 'objective/neutral'. The lexicon is automatically generated via a 4 step process [18]. As mentioned in the prior point, the POS of a term can influence its polarity shift, as such SentiWordNet 3.0 contains separate sentiment polarity values for different POS tags. To ensure the correct sentiment for a given term is retrieved, the inclusion of the POS tag is also used to select the sentiment values.

The probability distribution of a term's sentiment is then combined to generate a 3-dimensional vector.

- **Polarity Shift Embedding:** Polarity shift, is the directional polarity of a given word. For example, the term 'good' has a normal polarity shift within the sentence, "I found the meal to be good". However, the same term can have an inverse polarity shift, if used in a negating sentence, "The meal was not good at all". As such, the inclusion of polarity shift is important for sentiment analysis because the presence of negation completely reverses the semantic meaning of a phrase. As such, the exclusion of polarity shift can result in misunderstanding of a given phrase. A rudimentary approach is utilized to calculate the polarity shift of a given word. Terms in a tweet are systematically read and the polarity is determined to change if a negation term (i.e., 'not') is witnessed. Therefore, polarity orientation is considered 'normal' up to a negation term. Orientation of subsequent terms are then considered 'inverse' until the end of the phrase or the presence another negation term.

In order to consolidate the different embedding layers, the output of each embedding is then concatenated to generate a single word vector representation. This concatenation operation allows a scalable preprocessing pipeline, that is capable of interchangeable embedding's in order to capture varying linguistic features.

After a d-dimensional vector is generated for each word, vectors are then concatenated together to produce a $n \times d$ matrix, where n is the number of terms in the tweet and d is the dimensional size of the vector. In order to normalize the data between tweets and allow the preprocessed data to be fed into varying neural network architectures, zero padding and truncation is performed on

the 0-axis of the matrix. That is to say, zero-padding is performed on the n dimension of the matrix, if n is less than a threshold value, m. Likewise, truncation is performed on the n dimension, if n

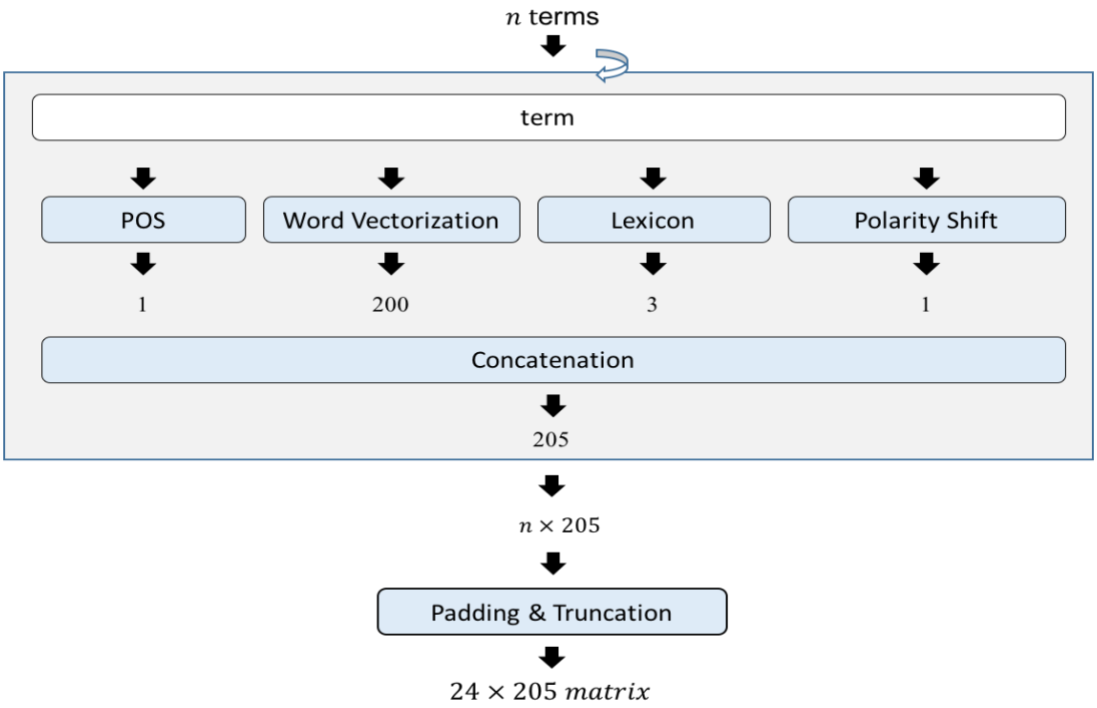


Figure 3.3: Illustration of block 3 and Dimensions of Preprocessing Pipeline.

is greater than m. As such, the padding and truncation operation outputs $m \times d$ matrix, where m is the max number of terms. In Fig 3.3, an outline of the dimensional output of each component within block 3 can be found. In the implementation of the ASAS for SRC, terms are embedded as 205-dimensional vectors and a tweets are padded or truncated to 24 terms. A max threshold of 24 was chosen based on the longest tweet within the dataset.

3.3 Deep Neural Networks

As mentioned earlier in section 2.2.2, the state-of-the-art sentiment classification is primarily conducted via deep neural networks. As such, in the current body of work, various neural network

architectures are empirically evaluated, selected and combined in an ensemble system. This section presents the varying neural network architecture and hyper parameters chosen for evaluation. The implementation and evaluation of the models in this thesis was conducted in a python environment utilizing the keras framework to develop the various models. A detailed list of the python packages used in this thesis is provided in Appendix A.

Due to the potential of each neural network model being incorporated within an ensemble system, there should be a common output among each neural network. As such, each neural networks' last layer contains a softmax layer with *K* number of neurons, where *K* is levels of sentiments (i.e., *K* = 3 for 'positive', 'negative', and 'neutral' sentiments). This provided a normalized probability distribution as output for all the neural networks. In addition, since various deep neural networks are all applied to learn the sentiment classification of SRC twitter data within three levels of sentiment, a weighted categorical cross-entropy is selected as a common loss function. Since there is an imbalance with the class distribution within the concussion dataset, a weighted loss function is used to force the model to focus more on the under sampled class. The combination of the weighted categorical cross-entropy and the common softmax output layer, allows a common learning objective among the neural network model.

In addition to the common loss function and output, other parameters which can be common among the neural network models, such as learning algorithm, epoch, and batch size are selected as followed. For learning algorithm, the Adam optimizer algorithm is selected to allow adaptive learning of the optimal weights [67]. Adam is a very popular optimizer that generates state-of-the-art performance due to its unique method of updating the weights of a neural network. Unlike, traditional stochastic gradient descent, which updates weight just by the current gradient, weight update in Adam is based on the corrected exponentially decaying weighted average of gradient

and square gradient. That is to say, Adam adaptively learns by placing more emphases on current gradient rather than past gradients as movement is built during each epoch. As per batch size and epoch size, a value of 100 and 40 is selected respectively. A high epoch size is selected to ensure all networks have adequate number of iterations to learn an optimal pattern from the data. However, a high epoch value comes with the risk of overfitting the model to the training data. As such, an early detection method is applied during training of all the neural network models to ensure models do not over fit. After each epoch, early detection reviews the performance of the neural network, if the network is seen not to improve based on a threshold, then training is terminated early. In the implementation of the current work, performance is measured as the training accuracy of the neural network, and a threshold of 0.002 for improvement is considered. Since, early detection is measured on performance, a lower threshold of 0.002 (0.2% increase) that non-significant increase in performance is overlooked. To ensure training is not terminated prematurely, early detection waits 10 epochs of non-improvement before terminating. Once terminated, the weights of the neural network revert back to the values from the epoch that produced the highest performance.

3.3.1 Hyper Parameters

In this section, the unique hyper parameters for each neural network architecture is provided and discussed.

A. FFNN

Four different fully connected neural network models with varying number of layers and neurons were evaluated within this body of work. In all four models, a leaky rectified linear unit (ReLu)

activation with an alpha of 0.1 was selected in favor of the more traditional ReLu activation. The motivation in selecting a small alpha value of 0.1 in this thesis, is to allow only a small leak of negative information through. This decision was in order to mitigate the potential of permanently disabling some neurons which is present within the ReLu activation. The repercussions of a neuron being permanently disabled means that associated weights connected to the neuron may no longer update, potentially hindering the training of the network [37].

In addition to the activation, all four models also implement a variation to the dropout regularization to alleviate the potential problem of overfitting, further referred to as decaying dropout. Unlike the traditional dropout, the rate of dropout is decayed after each iteration. The formula to update the drop rate can be written as

$$r^e = \max\left(0, 1 - \frac{e}{E}\right) * r^0 \tag{3.1}$$

where:

- e is the current epoch.
- E is number of epoch iterations.
- r^0 is the initial drop rate.
- r^e is the drop rate for epoch e.

The principle to the decaying rate in the dropout is to quickly push a subset of weights during the initial epoch iterations of training. As the gradient moves down towards the regional cavity of the optimal solution, the restriction on the network is loosen and more weights are more freely trained during each passing epoch. Since a rate decay's over time and the network consists of multiple fully connected layers, a high initial rate of 0.50 is chosen for all four models.

Lastly, the differentiating hyper parameters of the number of hidden layers and neurons for each of the four models is as followed:

1. A FFNN model containing two hidden layers each containing 400 neurons.

2. A FFNN model containing three hidden layers each with 400 neurons.

3. A FFNN model containing 4 hidden layers with (775, 225, 75, and 25) neurons from the input layer to the output layer, respectively.

4. A FFNN model containing 4 hidden layers with (400, 200, 100, and 50) neurons from the input layer to the output layer, respectively.

The motivation to the four varying network sizes, is to evaluate models with increasing complexity via the addition of hidden layers. In addition, a bottleneck like structure in the last two FFNN model with 4 hidden layers attempts to linearly down sample the size of the input to the size of the output. The hypothesis to this approach, is to allow the network layers to learn the latent pattern that would down sample the input to the output in the attempt to project/encode the input space onto the output.

B. CNN

Similar to FFNN, the CNNs implemented in this body of work, utilize leaky ReLu activation with an alpha value of 0.1 and a decaying dropout with a rate of 0.2 after the fully connected layer with 30 neurons. Since CNN performs feature extraction and down samples the input data, the required number of neurons in the fully connected layer is less. As such, the initial drop rate can be set

much lower. Since terms are embedded as 205-dimensional vectors, the number of filters per filter size is also 205, maintaining the vector dimension of each term.

As per the filter sizes of the CNN models, two separate approaches have been evaluated in this body of work. The first approach, contains a single layer with varying filter sizes that are then down sampled and concatenated to produce an extracted feature vector for the fully connected layer. An example of this approach is illustrated in Fig 3.4. It can be seen that varying filter sizes capture different n-gram representation of the input text and sampling only the most significant information from the n-gram mapping. By sampling only the most influential information from each n-gram mapping, the network is able to extract important n-gram features which can then be used within the classification layer.

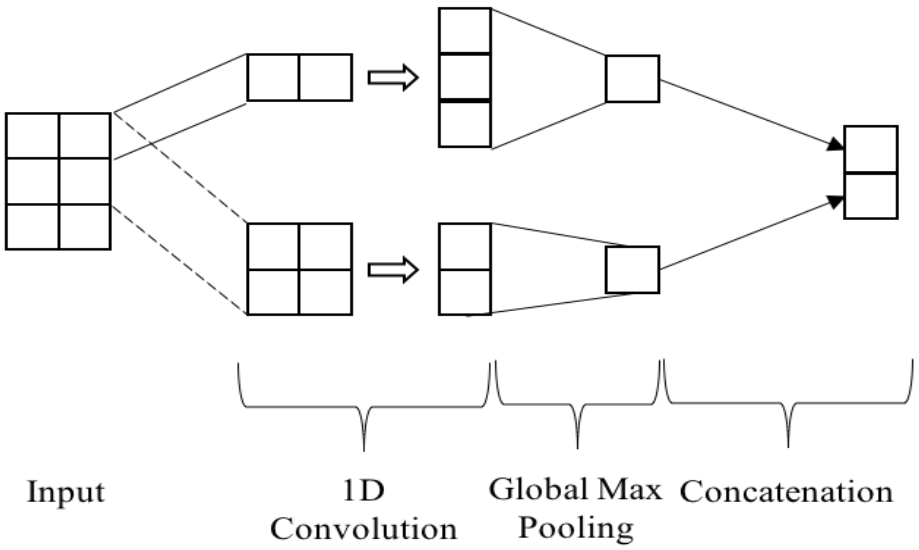


Figure 3.4: Diagram illustrating a single 1D CNN. The layer consists of 2 filters of size 1 and 2 respectively.

In contrast, rather than extending the number of filter sizes in a single layer. The second approach extends the number of filter sizes into two layers, such that the feature map from one convolution filter is feed into the next convolution filter. This 2 layer stacked 1D convolutional layer is further illustrated in Fig 3.5. The principle behind the second approach, aims to extract higher level features from the first layer n-gram. For example, given the input sentence 'Concussion is a serious matter' and a multilayer CNN with filter size of 2 for the first and second convolution layer. The first layer attempts to learn the 2-gram features of 'concussion is', 'is a', 'a serious', and 'serious matter'. By stacking a second convolution layer with filter size 2, the network will attempt to learn the 2-gram features of the 2-gram features already learned (i.e., it will attempt to learn high-level feature of 'concussion is a', 'is a serious', and so forth).

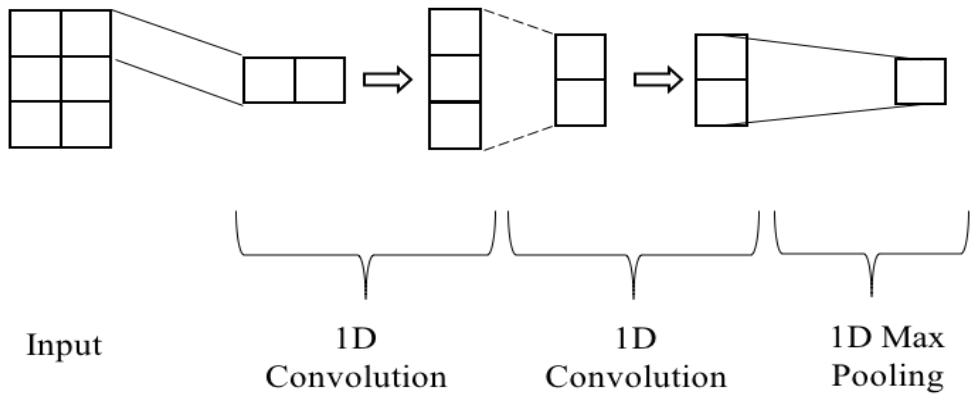


Figure 3.5: Diagram illustrating a 2 layer stack 1D convolution network. The first 1D convolution has a filter size of 3 generated a 1×3 *feature map. A second convolution takes the* 1×3 *feature map and applies a 1D convolution with filter size 2. 1D max Pooling is then applied after the second convolution producing a scaler value.*

Therefore, in order to determine suitable n-gram representation for the concussion dataset, varying models with different filter sizes are evaluated. Specifically, the following five CNN models with their filter size parameters were evaluated:

- A single layer 1D-CNN with filter sizes of (1, 2).
- A single layer 1D-CNN with filter sizes (1, 2, 3).
- A single layer 1D-CNN with filer sizes (3, 4, 5).
- A single layer 1D-CNN with filer sizes (1, 2, 3, 4, 5).
- A multi-layer 1D-CNN with filter sizes (1 and 2) for the first and second layers, respectively.

C. RNN

Since RNN's typically contain less tuning in regards to the required hyper parameters, only 3 varying RNN models are evaluated with the variation coming primarily from their architecture (LSTM, bidirectional LSTM, and GRU). A bidirectional LSTM, is included in the evaluation to determine if the directional orientation of the sequence provides significant impact to the performance. For example, both LSTM and GRU are only able to generate prediction based on the current and prior words. However, bidirectional LSTM also generate prediction based on the current and future words. The intuition to the bidirectional LSTM, is that in some cases the future words beside a current word is more significant than its prior neighboring words. Similar to CNN, all 3 RNN models contain a fully connected layer with 30 neurons for the sentiment classification section of the network. Again, a decaying dropout with an initial rate of 0.2 is applied after the

fully connected layer. Lastly since each term in the input is represented as a 205-dimensional vector, the size of each RNN model is 205 neurons.

D. TCN

Since the varying factors that influence TCN differently from the other neural network architecture (FFNN, RNN, and CNN) are the filter size and the total depth of the network, only a single TCN model is evaluated. Similar to the other models, we configure the TCN with a drop rate of 0.2 and modify the vanilla TCN slightly by replacing the standard ReLu activation function with a leaky ReLu (also configured with an alpha of 0.1, similar to the other evaluated models).

As per the filter size, a size of 3 was chosen due to of the characteristics of tweets being short sentences. With a filter size of 3, the TCN would require a total depth of 4 in order for the field of view to span across a whole input tweet. Since the dilation at a given layer can be calculated as $d^i = 2^i$, we would have dilation values of (1, 2, 4, 8) for each layer respectively. We can then calculate the effective history or the input scope of a layer to its preceding layer by calculating $eh^i = (k-1)d^i$, where k is the filter size, d is the dilation, i is the layer, and eh is the effective history [37]. As such, we get an effective history of (2, 4, 8, 16) at each subsequent layer. Finally, the field of view at a given layer can then be calculated as $\sum_{j=0}^{i} eh^j$, resulting in values of (2, 6, 14, 30) at each depth. Therefore, since the max input tweet is 24 terms, TCN with a depth of 4 is required, in order for the scope of the output to see the whole input data.

3.4 Transfer Learning

In this body of work, the approach to transfer learning within deep classification neural network deviate from conventional sense. It is very common in the research domain of deep neural network to apply state-of-the-art pre-trained deep neural network classification, such as Oxford's VGG, Goggle's Inception v3, or Microsoft's ResNet, to the specific domain problem. However, this approach, poses a few limitations and challenges within this body of work. First, publicly available state-of-the art pre-trained deep neural network models for sentiment analysis, is very scarce, since majority were developed for image processing. Secondly, the use of these state-of-the-art model often require a preprocessing that conforms the data to a specific form required by the pre-trained model. However, this negates the different embedding information captured by combining different embedding techniques and methods, since the pre-trained models where trained with only a single specific embedding.

Therefore, transfer learning approach is done by utilizing external datasets from a different domain problem, but with similar characteristics as that of the concussion dataset. Instead of leveraging the state-of-the-art pre-trained models, the five following models: LSTM, bidirectional LSTM, multi-layer CNN, single-layer CNN with filter sizes [1,2], and single-layer CNN with filter sizes [1,2,3], as stated in section 3.3, is pre-trained with the SemEval-2016 and Kaggle weather datasets prior to training with the concussion data. Since SemEval-2016 and Kaggle weather consist of tweeter data that have been labelled with the document-level sentiment of the tweet, only those 2 among the 6 external datasets are considered for pre-training. While, each dataset is focused on solving a different problem, they are all twitter data that illustrate a sentiment of a specific topic from the author. Thus each dataset has similar technical challenges, as per the length and informal

structure of the data. While nether the SemEval-2016 nor the Kaggle weather ranks the sentiment of two different topics, the hypothesis to this pre-training approach, is to allow the system to first learn the general understanding of a sentiment. Once the network learns the sentiment of a general tweet, we transfer that learned model on to the concussion dataset, such that the model now attempts to learn the ranking between the sentiments of different topics (i.e., concussion vs sports).

3.5 Ensemble

The last component of the automated sentiment analysis system, is the ensemble system that combines the classification of the neural network models to generate a consolidated classification. Again, the motivation towards the ensemble approach stems from the informal structure of tweets that allows varying writing styles. The idea, is to leverage the difference between the neural network architectures that cause the networks to focus on slightly different components during training. For example, by considering CNN architecture, the model is taking a more n-gram approach to training, whereas an LSTM is attempting to learn the sequential pattern via memory retention, or in the case of TCN that attempts to learn based on focusing on specific sections within a tweet. The principle towards this approach, while all models attempt to learn the pattern that maps the input to a sentiment, their different approach will capture information that may otherwise be missed by another model. However, too much varying representation can also be detrimental. Therefore, while multiple neural network models were evaluated, the classification votes of only the top performing neural networks are considered in the ensemble. Table 3.4 illustrates which of the evaluated neural network models were included in the ensemble.

Table 3.4: List indicating neural network models added to ensemble

Models	No Pre-training	Pre-training w/ SemEval-2016	Pre-training w/ Kaggle Weather
LSTM	✓	✓	
Multi-layer CNN	✓	✓	
Single layer CNN w/ Filter [1,2]	✓		
Single layer CNN w/ Filter [1,2,3]	✓	✓	

Traditional ensemble approaches focus on applying a 'hard' or 'soft' voting system to consolidate the output of different models to produce a majority vote. In the case of 'hard' voting, the concrete classification (i.e., 'positive', 'negative', or 'neutral') is taken into considered, whereas 'soft' voting considered the probability distribution of each class (i.e., how confident the model perceives the tweet as being 'positive', 'negative', or 'neutral'). However, in either cases, each model has equal influence towards the final decision. Intuitively, one can argue that in cases where experience and knowledge influences the correct the decision, the weight of each vote should be treated differently. For example, in the case of a model, when voting class, A is 80% correct but when voting class B is only 5% correct of the time. Therefore, in a situation such as this, the model has high precision for class A but low precision with class B. Therefore, when consider the vote from the model, the model's vote would be considered more its vote is A but its influence will be much lower when its vote is B. Based on the principle that the votes should be weighted depending on the situation, this body of work deviates from the standard ensemble approach.

Instead, the ensemble system is implemented by a FFNN that learns the influential pattern that each model has towards the final decision. Since varying influence from the varying votes from the model can lead to the ideal decision, the complexity of the pattern could present a non-linear problem, therefore a FFNN is implemented for the ensemble system. Since the input of the FFNN

are simply the votes from the 4 models, a shallow network with 2 layers with a bottleneck structure is implemented. Again, the principle to the bottleneck approach, is for each layer to learn the non-linear projection of the subsequent layer to a down sampled size until finally projecting onto the 3 level classification. Specifically, two FFNN models were evaluated in this body of work. The first network contained 21 and 7 neurons for the first and second layer, respectively. Lastly, the second larger network, contained 30 and 18 neurons.

3.6 Summary

To summarize, the proposed system contains 3 main components: the preprocessing pipeline, neural network models, and the ensemble.

The preprocessing pipeline, contains 3 blocks which cleans, normalize and generates a vector representation per word. The initial cleaning block performance standard natural language preprocessing algorithms that encodes related terms, identifies and segments hashtags, and shorten elongated words. The data is then normalized to mitigate the sparsity within the data. This is conducted via tokenization, lemmatization, and NER. Lastly, all highly frequent terms are discarded prior to vectorization. An important component to the preprocessing pipeline is the vectorization block. This block allows different embedding algorithms (word vectorization, part-of-speech embedding, lexicon embedding, and polarity shift embedding) to be combined such that different linguistic features are extracted and concatenated.

The second component are the neural network models, which learns the relational pattern between the sample twitter data with its sentiment. Since varying neural network architecture are available and have demonstrated to be effective in natural language processing (NLP) in recent years, this

thesis exams a variety of neural network models to be selected for the ensemble system. Specifically, a total of 13 different models are evaluated (four FFNNs, five CNNs, three RNNs, and one TCN).

Through examination, a total of seven models are selected for the ensemble system:

- Non pre-trained LSTM
- Non pre-trained multi-layer CNN
- Non pre-trained single layer CNN with filter sizes [1, 2]
- Non pre-trained single layer CNN with filter sizes [1, 2, 3]
- LSTM pre-trained on the SemEval-2016 dataset
- Multi-layer CNN pre-trained on SemEval-2016 dataset
- Single layer CNN with filter sizes [1, 2, 3] pre-trained on the SemEval-2016 dataset.

4 Experiments & Results

This chapter presents an empirical evaluation of the various neural network models and the proposed ensemble SA system. The first section provides an overview of the metrics used. The second section presents and discussed the evaluation of the individual neural network models. Lastly, the third section presents and discussed the evolution of the pre-training and the final ensemble system.

4.1 Metrics

The problem of predicting the sentiment of a tweet lies in the subset of classification problems within machine learning. This body of work's evaluation is primarily based on the true positive, false positive, true negative, and false negative metrics. It should, however, be clarified that the use of the terms 'positive' and 'negative' in this discussion of metrics, is not in reference to positive and negative sentiments of tweets, rather it is related to the classification of data. The terms positive and negative, in this section, shall thus refer to a data point belonging or not belonging to a specific class.

Fig. 4.1 illustrates how true positives represent predictions that were correctly predicted as belonging to the specific class (i.e., class 'A' in the diagram). Similarly, true negatives are predictions that were correctly predicted as not-belonging to the specific class. On the other hand, false positive and false negative indicate predictions that were incorrectly predicted as belonging or not-belonging to the class, respectively. Any other use of the term throughout this section shall be explicitly stated.

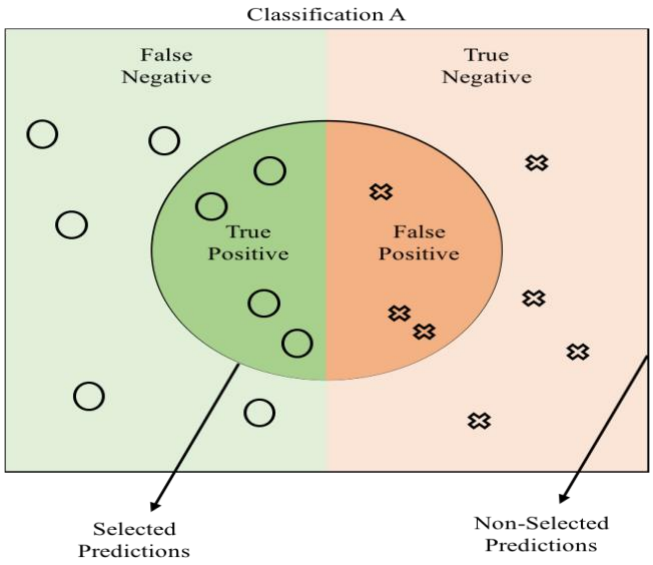


Figure 4.1: Diagram illustrating the metrics of true/false positive and true/false negative for a specific label 'A'. The circular region indicates data points classified as 'A' (Selected Predictions). The non-selected predictions (regain within the box but not in the circle) are data points not predicted as 'A'. Circular data points, represent ground-truth labels of 'A' and the X points represent ground-truth labels of non 'A'.

Precision is used to evaluate the rate at which a model correctly selected tweets that actually belong to a given sentiment. The precision of a given sentiment can be formally written as

$$Precision^{s} = \frac{TP^{s}}{TP^{s} + FP^{s}} \tag{4.1}$$

where:

- s is given the sentiment (positive, negative, and neutral).
- TP is the true positive of the sentiment.
- FP is the false positive of the sentiment.

Since precision is calculated based on true positive with false positive, a low precision score would indicate that the model is overly predicting a class to belong to a specific sentiment. Thus producing a model with a high ratio of false positives. From (4.1), the precision of each sentiment can be averaged and written as followed to provide the overall precision of a model:

$$Precision = \sum_{s} \left(\frac{n^s}{N}\right) Precision^s \tag{4.2}$$

where:

- s *is given the sentiment (positive, negative, and neutral).*
- n^s *is the total number of* s *labelled tweets.*
- N *is the total number of tweets.*

As mentioned earlier, the concussion dataset contains an unequal distribution of sampled sentiments, which causes misrepresentation for the sentiments. While this is accounted in the implementation of each neural network via the weighted loss function, the misrepresentation is only addressed during training and not during evaluation. As such, the overall precision of a model is averaged based on the weighted representation of the sentiment rather can a macro average.

Another important metric that is considered in this body of work is recall. While precision illustrates how accepting a model is at predicting data as positive, recall on the other hand can be viewed as how restrictive a model is at accepting a data as being positive. Therefore, a model with low recall can be considered restrictive, resulting in a high ratio of false negatives. Similar to precision, the overall recall score of a model can be formulated as

$$Recall^s = \frac{TP^s}{TP^s + FN^s} \tag{4.3}$$

$$Recall = \sum_s \left(\frac{n^s}{N}\right) Recall^s \tag{4.4}$$

where:

- s is given the sentiment (positive, negative and neutral).
- TP^s is the true positive of the sentiment.
- FN^s is the false positive of the sentiment.
- n^s is the total number of s labelled tweets.
- *N is the total number of tweets.*

While both recall and precision provide important information on the predictive behaviour of a model, the final comparison conducted on the performance of each model is measured with an F1-score. In determining the models that are used within the ensemble system, the F1-score is used in the evaluation. The F1-score is measured based on the recall and precision, providing the harmonic average between the two and be calculated as

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.5}$$

where:

- *Precision* is the weighted precision score for the three sentiments (positive, negative, and neutral) from (4.2)
- *Recall is the weighted recall score for the three sentiments (positive, negative, and neutral) from (4.4)*

Since both precision and recall provide a value between zero to one, we can adjust the F1-score to maintain the same range by multiplying it by 2, as shown in the (4.5). This allows us to interpret a high F1-score, towards one, as a model containing a high average between precision and recall. Vise-versa, a low F1-score, towards zero, would indicate a lower performing model with low precision and recall average.

Since different external datasets are used to evaluate the different neural network models, and each of the external dataset contain their own common metrics for evaluation, an additional metric is used primarily as a comparison with other state-of-the-art models. In this body of work, accuracy is used as a secondary metric to evaluate the varying models with that of other state-of-the-art models. Accuracy is a metric traditionally used to simply get the ratio of correctly classified predictions and can be calculated as

$$accuracy = \frac{1}{N}\sum_{i}^{N}\begin{cases}1, & if\ (y_i' = y_i)\\ 0, & if\ (y_i' \neq y_i)\end{cases} \tag{4.5}$$

where:

- N *is the total number of samples.*
- y_i' *is the prediction for the* i^{th} *sample.*
- y_i *is the ground-truth of the* i^{th} *sample.*

4.2 Neural Network Results

The discussion here is presented in three different sections, presenting different analysis conducted on the individual neural network models with the varying datasets. The first analysis, presents the comparative results of the varying neural network models evaluated in this body of work with that of other state-of-the-art models used in the same dataset. The second analysis, presents the comparison of the varying neural network models among themselves and illustrates the subset of optimal neural network architectures for sentiment analysis of SRC. Lastly, an analysis conducted on the unbalance sampling of sentiment within the concussion dataset is presented.

The standard 80/20 percent split for training and testing data was performed on all the experiments in the following section. Since the initial weights of a neural network contribute to the final performance of the model, each experiment was conducted 30 times with a newly randomized initial weight, with the highest performing iteration being recorded. A large iteration size of 30, allows for a good sample of different initial weights to be tested. In addition, larger sampling of different initial weights larger than 30 did not yield significant improvement. Lastly, among all the experiments, the optimization of all the models used the F1-score as its primary metrics.

4.2.1 Analysis 1

In this analysis, the accuracy of each model is compared with other systems proposed in related works that have been evaluated on the same dataset. Specifically, a comparison is conducted on the semeval-2016 and senti-target datasets, since both naturally consist of 3-level sentiments and were both evaluated with the common accuracy metric. While other datasets were included in the current body of work, they were excluded in the comparison for the following reasons:

- **Kaggle Weather:** The performance of other external models is measured based on the root mean square error (RMSE) due to the original nature of the competition. In addition to sentiment classification, the Kaggle weather contains other classification tags that are measured in the performance. Since the focus of our work is SA, evaluating external models that also predict the additional classification labels with only models that predict the sentiment classification cannot be justified.

- **Rotten Tomato**: Similar to Kaggle weather, the original dataset is composed of 5-level SA and the body of work is focus on the classification of sentiments in only 3-levels. A comparison between the models presented in this body of work and the external models would favour the simplified problem domain. In this case, the models presented in the current work would demonstrate higher performance due to the reduced level in the sentiments.

- **UCI:** The dataset size for UCI is very small and typically deep neural networks benefit from, and require a large training sample to produce state-of-the-art performance. Thus, the limiting dataset size would favour more traditional machine learning approaches, such as regression based, instead of other more complex models that produce state-of-the-art performance on the same problem domain but with larger sample size. In addition, no other body of work is evaluated on the UCI dataset, aside from the original source [59], which conducts an evaluation on each individual review site, reducing the sample size even further.

- **UMich650:** Similar to UCI, UMich650 is also a very small dataset, which could again lead to favouring more traditional machine learning approaches. In addition, the UMich650 is

retrieved via the private Kaggle competition hosted by the University of Michigan that does not support further evaluation for comparison.

Table 4.1, illustrates that five neural networks presented in the current body of work rank in the top 10 in accuracy among other state-of-the-art systems. Interestingly, the models that appear to be comparable with other state-of-the-art systems use the same neural network architectures (i.e., single layer CNN, multi-layer CNN, LSTM, and bidirectional LSTM) that commonly have been used in recent years for SA and natural language processing within different domain problems. Another observation one may gather from the result, is the filter size of the single layer CNN's which demonstrated better performance. Both single layer CNN models, consist of small window sizes of [1, 2] and [1, 2, 3], correlating to feature extraction of low n-gram representation of 1, 2, and 3. This small n-gram/filter size dependency may be explained by the small document size of tweets. Large n-gram representation could be introducing additional noise in the problem domain of short documents.

Table 4.2, illustrates that the presented models do not show comparable performance on the senti-target dataset as it did with the semeval-2016. This lack of performance may stem from nature of the dataset geared towards target-level SA. Since the additional target-topic input is excluded in this body of work, the vector embedding produced by the preprocessing pipeline contains no information on the target. This missing embedding of the target-topic can explain the performance produced by the varying neural network model. Since the networks are not provided with the specific target of interest, they are also tasked to implicitly learn the target of interest pertaining to the tweet and sentiment. In conclusion, while the specific configuration of the preprocessing pipeline may yield comparable results to state-of-the-art systems for document-level SA, additional tuning or extension is required for target-level SA.

Table 4.1: Accuracy results semeval-2016 dataset. Results of the external systems from the Semeval-2016: Task A competition are compared with the results of the neural network models presented in the current body of work. While a total of 34 systems contributed to the competition, only the top 15 rank systems are illustrated in descending order of accuracy. [48]

MODEL	***LAYER/FILTER SIZE***	***ACCURACY***
FFNN	*[400, 400]*	*0.595*
	[400, 400, 400]	*0.590*
	[775, 225, 75, 25]	*0.588*
SINGLE LAYER CNN	*[1, 2]*	***0.617***
	[1, 2, 3]	***0.611***
	[3, 4, 5]	*0.602*
	[1, 2, 3, 4, 5]	*0.606*
MULTI-LAYER CNN	*[1, 2]*	***0.622***
GRU	*[205]*	*0.606*
LSTM	*[205]*	***0.612***
BI-DIR LSTM	*[205]*	***0.622***
TCN	*[3]*	*0.596*
EXTERNAL SYSTEM		***ACCURACY***
SWISSCHESSE		*0.646*
NTNUSENTEVAL		*0.643*
UNIPI		*0.639*
CUFE		*0.637*
INSIGHT-1		*0.635*
AUEB.TWITTER.SENTIMENT		*0.629*
SENSEI-LIF		*0.617*
UNIMELB		*0.616*
SENTI-SYS		*0.609*
INESC-ID		*0.600*
THUIR		*0.596*
I2RNTU		*0.593*
LYS		*0.585*
PUT		*0.584*
UOFL		*0.572*
BASELINE		***0.342***

Table 4.2: Accuracy results of senti-target. Results from other state-of-the-art systems illustrated in the works of [58] are compared with the results of the neural network models presented in the current body of work.

MODEL	***LAYER/FILTER SIZE***	***ACCURACY***
FFNN	*[400, 400]*	*0.595*
	[400, 400, 400]	*0.590*
	[775, 225, 75, 25]	*0.588*
SINGLE LAYER CNN	*[1, 2]*	***0.617***
	[1, 2, 3]	***0.611***
	[3, 4, 5]	*0.602*
	[1, 2, 3, 4, 5]	*0.606*
MULTI-LAYER CNN	*[1, 2]*	***0.622***
GRU	*[205]*	*0.606*
LSTM	*[205]*	***0.612***
BI-DIR LSTM	*[205]*	***0.622***
TCN	*[3]*	*0.596*
APPROACH		***ACCURACY***
ADARNN-COMB		*0.663*
ADARNN-W/E		*0.658*
ADARNN-W/OE		*0.649*
SVM-DEP		*0.634*
RNN		*0.630*
SVM-INDEP		*0.627*
SVM-CONN		*0.600*

4.2.2 Analysis 2

A total of 12 models were evaluated on each of the 6 external datasets and the primary sports related concussion dataset, resulting in 72 experiments. The models were also evaluated on the 6 external datasets, to further analyse the adaptability of each of the models to different datasets with various domain problems. Among the proposed models in this thesis, this analysis determines the top models, which still perform well regardless of the problem domain. Table 4.3 illustrates the F1

results for each of the experiments, with top 5 models for each dataset highlighted. Additional material on the analysis results of these models can also be located in Appendix C.

Table 4.3: Results of F1-score on all datasets: Sports related concussion (SRC), semeval-2016 (SEMEVAL), Kaggle weather (KG), rotten tomato (RT), senti-target, uci, umich650.

MODEL	***LAYER/ FILTER SIZES***	***SRC***	***SEM-EVAL***	***KG***	***RT***	***SENTI-TARGET***	***UCI***	***UMICH 650***
FFNN	*[400, 400]*	*0.5529*	*0.5886*	*0.9043*	*0.8312*	*0.5744*	*0.7894*	*0.8693*
	[400, 400, 400]	*0.5557*	*0.5877*	*0.9053*	*0.8300*	*0.5736*	*0.7887*	*0.8696*
	[775, 225, 75, 25]	*0.5500*	*0.5737*	*0.9069*	*0.8238*	*0.5662*	-	-
	[400, 200, 100, 50]	-	-	-	-	-	*0.8021*	*0.8692*
SINGLE LAYER CNN	*[1, 2]*	***0.6201***	***0.6230***	***0.9324***	***0.8734***	***0.6131***	***0.8504***	***0.8937***
	[1, 2, 3]	***0.6135***	***0.6159***	***0.9346***	***0.8772***	***0.6214***	*0.8445*	*0.8887*
	[3, 4, 5]	*0.5983*	*0.6082*	*0.9281*	*0.8709*	***0.6196***	*0.8367*	*0.8933*
	[1, 2, 3, 4, 5]	***0.6122***	*0.6099*	*0.9322*	***0.8751***	***0.6213***	***0.8487***	***0.9032***
MULTI-LAYER CNN	*[1, 2]*	***0.6121***	***0.6224***	*0.9319*	***0.8739***	*0.6086*	***0.8498***	***0.8933***
GRU	*[205]*	*0.6087*	*0.6082*	***0.9325***	*0.8665*	*0.5889*	***0.8470***	***0.8938***
LSTM	*[205]*	***0.6138***	***0.6137***	***0.9325***	*0.8718*	*0.6040*	***0.8502***	*0.8889*
BI-DIR LSTM	*[205]*	*0.6078*	***0.6208***	***0.9328***	***0.8730***	***0.6140***	*0.8410*	*0.8790*
TCN	*[3]*	*0.5615*	*0.5962*	*0.9290*	*0.8558*	*0.5909*	*0.8396*	***0.8984***

Based on the results of each individual model, the single layer CNN with filter sizes [1, 2] shows consistently high performance across all dataset. In addition, to the filter sizes of [1, 2], the combination of filter sizes [1, 2, 3] and [1, 2, 3, 4, 5] for a single layer CNN also show high performance among some of the datasets. However, the single layer CNN with filter sizes [3, 4, 5], does not appear to rank among the top performing models. This would suggest that the

information gathered from higher order filter sizes provide less to no additional information than smaller filter sizes. As such, it can be concluded that the performance of the single layer CNN with filter sizes [1, 2, 3, 4, 5] can be primarily contributed by the first 3 smaller sizes [1, 2, 3]. The lack of benefit from larger filter sizes may be explained by the small document size of a tweet. Since a tweet only contains a small number of terms, smaller n-gram representation would provide more information than larger n-gram. A smaller n-gram or smaller filter size would provide a smaller window scope, thus extracting finer-details. In contrast, the larger filter size, would contain a larger window scope, causing a broader representation of more terms. Therefore, we can treat the single layer CNN with filters [1, 2, 3, 4, 5] as redundant and only maintain the single layer CNN models with filter sizes [1, 2] and [1, 2, 3].

In addition to the single layer CNN, the multi-layer CNN also ranks in the top 5 among 5 of the 7 datasets. As illustrated in Table 4.3, one can see that the performance of the multi-layer CNN does not deviate from its single layer counter-part. This would suggest that the n-gram approach of feature extraction yields promising performance in SA. Specifically, a small n-gram is more beneficial than higher n-gram values in the analysis of social media messages like twitter, which are characteristically very brief sentences. This would suggest that a small window scope that focus on immediate neighbouring words is an important feature in the SA of brief sentences. Therefore, the multi-layer CNN model is also considered for further evaluation.

Interestingly, in all datasets, all feed-forward neural network models significantly underperform with respect to the other models. This could be explained by the sequential characteristic of our problem. Since FFNN does not consider the order or position of the input, we may conclude that the correlation of a term to other terms is significant in the interoperation of a tweet's sentiment. This intuitively makes sense. If we take a sentence and scramble the words, it becomes very

difficult for one to determine the sentiment of the sentence due to the grammatical structure of the sentence being destroyed. While it may be possible to guess the sentiment of sentences with only a single topic primarily based just on the appearance of positive and negative terms. This task becomes significantly difficult once multiple subjects are discussed within the sentence. When multiple subjects are introduced, identifying the association between sentimental terms and the subject they are referencing become much more important. However, removing the sequential order in which the terms appear in the sentence, increases the complexity of the task. Therefore, all feed-forward models are excluded from further evaluations.

As per the neural network models with a sequential depend approach, which primarily focus on the sequential order of the inputs. Both LSTM and bidirectional LSTM perform equally well for the varying datasets. On the other hand, both TCN and GRU show less desirable performance compared to LSTM models. Since GRU showed good performance in the UCI and UMich650, it can be argued that GRU may be more suitable for problems with a smaller data sample compared to LSTM. The added complexity of maintaining the memory cell in LSTM as opposed to the simplified gating mechanics in GRU, could account for this discrepancy. A more interesting observation from Table 4.3 is the underperformance of TCN model, aside from the UMich650 dataset. This observation can lead to the conclusion that the regional attention is a less important feature in the domain of SA than what we considered in our original hypothesis. Another possible explanation for the lack of performance is a potential limitation in the fixed history scope of TCN. Unlike RNN type networks, that try to learn the amount of information to retain forcing the network to understand the short-term vs long-term history dependency of the input. However, TCN contains a fixed history scope via the hyper parameters: filter size and network depth. Therefore, the length of the history is pre-determined prior to training.

In summary based on the performance evaluation conducted of each dataset, the neural network models considered for ensemble are reduced to the following subset:

- Single layer CNN with filter sizes [1, 2]
- Single layer CNN with filter sizes [1, 2, 3]
- Multi-layer CNN with filter sizes [1, 2]
- LSTM
- Bidirectional LSTM

4.2.3 Analysis 3

The next set of experiments were conducted to analyse and evaluate the performance of the models given the imbalance sampling of the sentiments. Traditionally, a dataset with a balance proportion of classes is the ideal situation for training. The proportional dataset will allow the model to see an equal number of samples from each classes. This prevents the model from over compensating for classes that are over sampled or undercompensating for classes that are under sampled. As discussed in section 3.1, the distribution of the sentiments in the concussion datasets is imbalanced. Since increasing the dataset to establish a balance distribution of sentiment was not feasible due to time and human resource burden associated to manual labelling. Two methods of discarding tweets were utilized in this analysis to generate two proportional distributions between the sentiments (‘positive’, ‘negative’, and ‘neutral’). In the first approach, a set of randomly selected ‘positive’ tweets were discarded such that the total number of ‘positive’ tweets equaled in the total number of ‘negative’ tweets. This resulted in discarding 4,726 random ‘positive’ tweets, producing a semi-proportional concussion dataset with 11,074 tweets. In the second approach, in addition to discarding ‘positive’ tweets. A set of randomly selected ‘neutral’ tweets where also discarded such

that the total number of tweets for each sentiment (‘positive’, ‘negative’, and ‘neutral’) were equal. This second approach, generated a fully-proportional concussion dataset with 8,190 tweets by discarding an additional 2,884 tweets.

Fig. 4.2 compares the performance of the original non-proportional concussion data with the two generated semi- and fully- proportional dataset. While the performance of the adjusted dataset underperformed the non-proportional data. One can observe the relation between the distribution of the concussion dataset and the increase of sample size. The observation suggest that maintaining a proportional distribution between ‘positive’ and ‘negative’ sentiments as the sample size increase is important to provide significant increase in performance by the increase in data. Additional analysis graphs are also available in Appendix D.

Interestingly, the removal of 4,726 ‘positive’ tweets that resulted in the number of ‘positive’ and ‘negative’ sentiments to be proportional, only exhibited a small degradation in performance. In contrast, the remove of 2,884 ‘neutral’ tweets exhibited larger degradation from the semi-proportional dataset. This observation may lead to the conclusion that the increase of ‘positive’ sentiments while not maintaining a proportional balance with ‘negative’ sentiments yield little increase in performance. In contrast, the maintenance of ‘neutral’ sentiments with the other two sentiments is less vital when increasing the sample size of the dataset. Therefore, the scaling of training samples should maintain a balance sample between ‘positive’ and ‘negative’ tweets to yield significance improvement in performance.

However, since the semi-proportional and fully-proportional distributions yielded worst performance, the utilization of all samples is a more important factor than the distribution of the sentiments in our case. As such in the case of the concussion dataset, one can conclude that the

benefit of a proportional distribution does not outweigh the repercussions of a reduced sample size. Therefore, in this thesis, the non-proportional concussion dataset is kept in order to maintain the largest sample size.

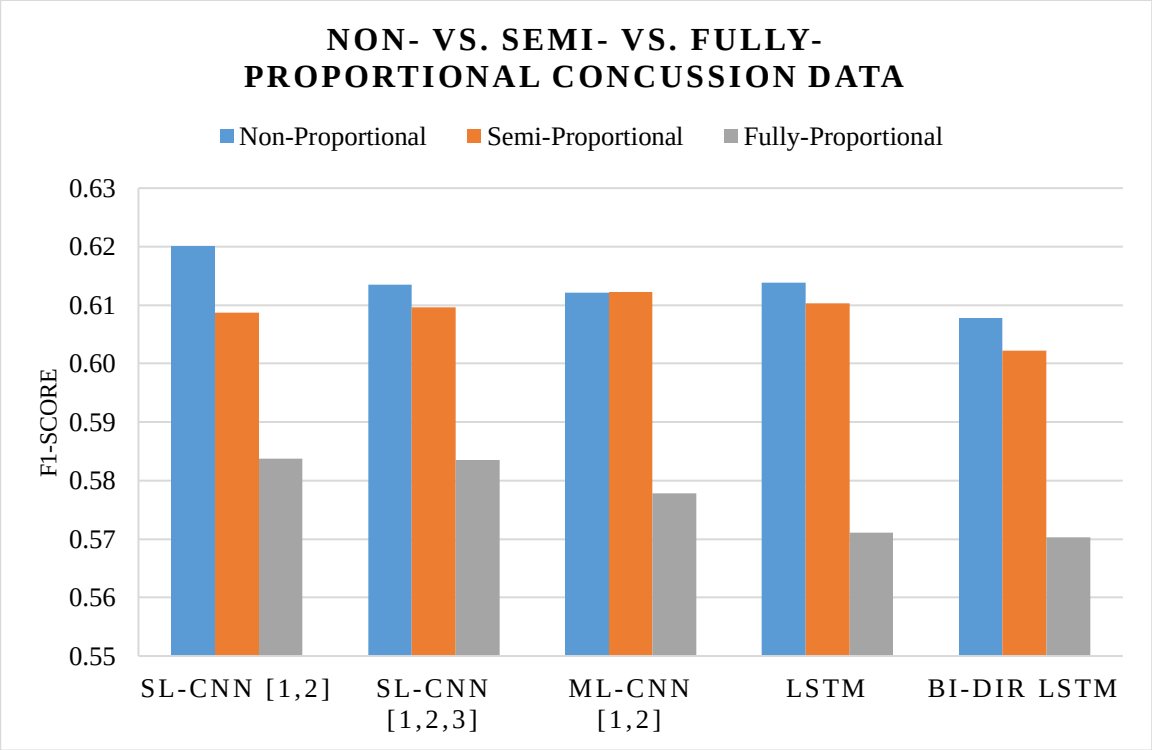


Figure 4.2: Line graph illustrating the F1-score for each model on the 3 proportional (non-, semi-, fully-) concussion dataset.

4.3 Pre-training & Ensemble Results

In this section, the results and evaluation of pre-training the presented neural network models from the previous section is first presented followed by the final performance of the ensemble system.

In this thesis, pre-training is conducted on the subset of five neural network architectures selected based on the analysis conducted in section 4.2.2. The models pre-trained on the semeval-2016 and Kaggle weather are further trained with the original non-proportional concussion dataset.

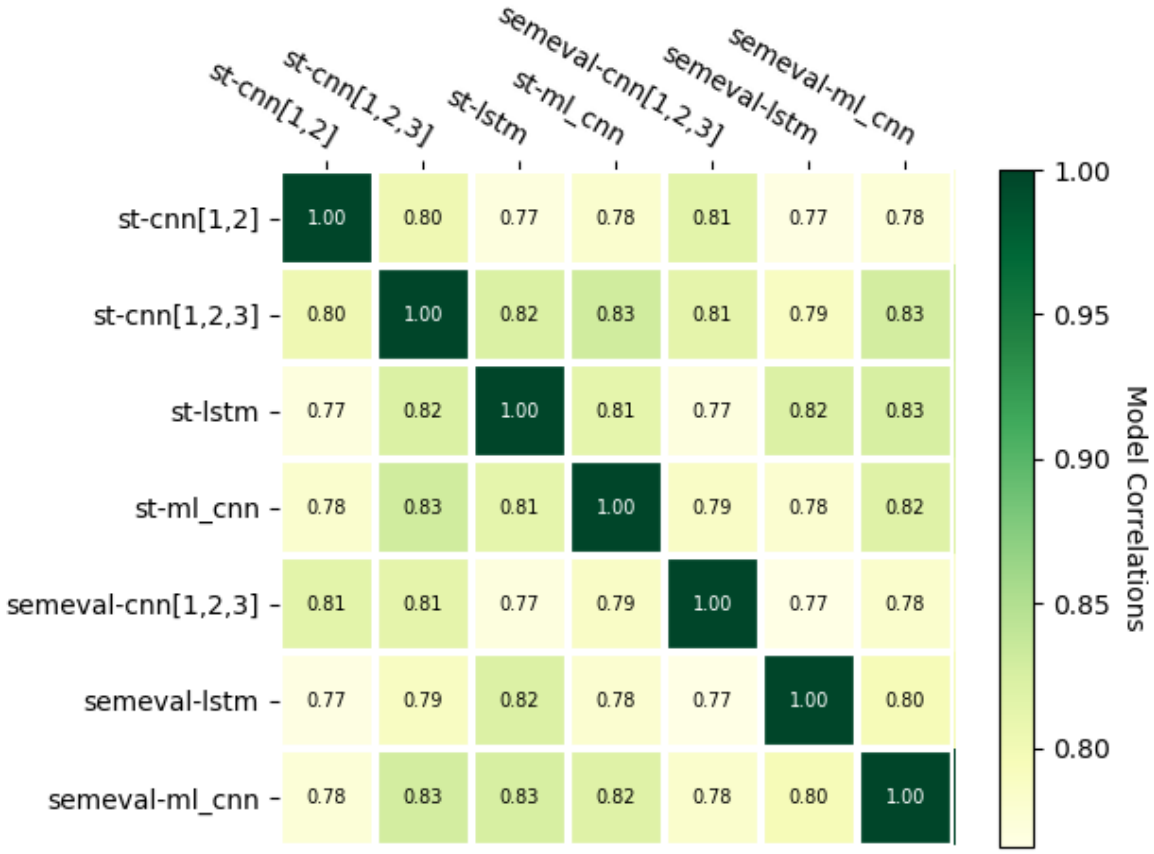


Figure 4.3: Heat map illustrating the correlation of the models' predictions.

An ensemble system is used due to the correlation of the models' prediction, as illustrated in Fig. 4.3. While the models produce similar performance score as outline in Table 4.4, the correlation between each model are not high (below 95% standard), which would indicate that the models are generating different predictions for some of the datasets. Therefore, it can be concluded that the models are capturing different information during training. This can potentially be explained by

the various architectures and parameters used in this thesis. This intuitively, motivated us to stack models so that the various predictions from the different models can be combined.

Based on the results illustrated in Table 4.4, it can be concluded that pre-training the model with the Kaggle weather provides a decrease in performance across the varying neural network models. This could be explained by the difference in the domain problem. The Kaggle weather primarily consists of sentiments related to the weather. Therefore, the similarly in the domain may be too little to produce any increase in performance. This illustrates that pre-training models on datasets with similar technical characteristic (i.e., unstructured, short character length) may not necessarily yield better performance. Therefore, experiment and evaluation should be conducted to determine the optimal models. As for semeval-2016, there are indication of similar or marginal improvement. This could be caused by some of the similar characters between the datasets. Both datasets contain an non proportional distribution of sentiments. In addition, preprocessing initially conducted on the SemEval-2016 by the authors of the dataset is very minimal, producing similar text as the raw tweets gather in this body of work. Also based on included perfromance of the single layer cnn with filter size (1, 2, 3) and LSTM models pre-trained on the semeval-2016 dataset, one can concluded that the bidirectional LSTM no longer compares with the other models. Therefore, the models can be further reduced by eliminating the bidirectional LSTM from the ensemble system.

Table 4.4: F1 results of models pre-trained on semeval-2016 and Kaggle weather (KG) trained on the concussion dataset. The original performance of the not pre-trained model is also illustrated for easy comparison. In addition, the top performing models are bolded.

MODEL	***LAYER/FILTER SIZES***	***NON PRE-TRAINED***	***PRE-TRAINED SEMEVAL-2016***	***PRE-TRAINED KG***
SINGLE LAYER CNN	*[1, 2]*	***0.6201***	*0.6076*	*0.6082*
	[1, 2, 3]	***0.6135***	***0.6139***	*0.6072*
MULTI-LAYER CNN	*[1, 2]*	***0.6121***	***0.6093***	*0.5994*
LSTM	*[205]*	***0.6138***	***0.6117***	*0.6084*
BI-DIR LSTM	*[205]*	*0.6078*	*0.6066*	*0.6087*

In order to determine the models to be selected for the ensemble system, models were selected based on a threshold. This threshold is calculated based on the average performance of the non pre-trained models was 0.613 ± 0.004. Therefore, a threshold of 0.609 was used to evaluate models for the ensemble system. In conclusion, with the exclusion of models pre-trained with the Kaggle weather dataset, bidirectional LSTM models, and other models pre-trained on the SemEval-2016 dataset, the following top performing models are selected for the ensemble system:

- Single layer CNN with filter sizes [1, 2], that was not pre-trained
- Single layer CNN with filter sizes [1, 2, 3], that was not pre-trained
- Multi-layer CNN that was not pre-trained
- LSTM that was not pre-trained
- Single layer CNN with filter sizes [1, 2, 3] that was pre-trained on the SemEval-2016 dataset
- Multi-layer CNN that was pre-trained on SemEval-2016 dataset
- LSTM that was pre-trained on the SemEval-2016 dataset

Table 4.5: Results of the ensemble system.

SYSTEM	*F1-SCORE*	*PRECISION*	*RECALL*	*ACCURACY*
ENSEMBLE	*0.6271*	*0.6271*	*0.6272*	*0.6272*

Lastly, the ensemble system was trained and tested with the above selected neural network models. In this body of work, 'hard' voting approach was used over 'soft' voting, in the aim of reducing noisy input for the FFNN ensemble system. For example, rather than introducing additional noise from less probably sentiments in a vote, the FFNN ensemble system shall only consider the highest probably sentiment in a vote and ignore all other probability distribution. Table 4.5 illustrates a slight increase in performance with the ensemble system as opposed to the prediction of an individual neural network model, in our case a F1-score of 0.6271.

4.4 Summary

To recap a total of five experiments were conducted for this thesis. The first experiments evaluated the proposed models presented in this thesis with other proposed models from other related works. The results indicated that five of the proposed neural network models on their own demonstrated results that compared well to other related works on the SemEval-2016 and Senti-Target dataset, only deviating 2.5% accuracy from the other systems. Afterwards, an experiment was conducted to evaluate the performance of each proposed model to one another. Each model was trained and evaluated on each of the six external datasets. The results of the experiment correlated with results illustrated in the first experiment. The top performing models among the six external datasets were also the same set that demonstrated high performance during the first evaluation. The results conclude that the following models are the optimal subset:

- Single layer CNN with filter sizes [1, 2]
- Single layer CNN with filter sizes [1, 2, 3]
- Multi-layer CNN with filter sizes [1, 2]
- LSTM
- Bidirectional LSTM

The third experiment was conducted to analyze the relationship with the distribution of the training sample and its effect on performance. Two balancing methods were evaluated in this experiment: semi-proportional (proportional distribution between ‘positive’ and ‘negative’ tweets) and fully-proportional (proportional distribution among all three sentiments). The results indicate that maintaining a semi-proportional distribution is important when scaling training samples. When increasing training data to increase performance, it is important to maintain an equal distribution between ‘positive’ and ‘negative’ tweets. The experiment shows that an increase in only ‘positive’ tweets does not necessary increase the overall performance of the model.

The forth experiment was conducted to analyze the effect of pre-training the models with dataset that contained similar characteristics but of a different subject. Models were first pre-trained on the twitter data from SemEval-2016 and Senti-Target prior to being training on the concussion dataset. While some models illustrated better performance on the SemEval-2016 dataset, the experiment indicated that datasets with similar characteristics (short informal unstructured posts) may not necessary yield better performance. This would indicate that evaluation should still be conducted when pre-training on datasets with similar characteristics but with different subject matters.

Lastly, an evaluation on the ensemble system using the subset of the seven optimal models was evaluated. The FFNN ensemble system generated an F1-score of 0.6271.

5 Conclusion

This thesis presents a body of work related to the development of an automated system to analyze twitter data for the purpose of understanding the general public's opinion of sports related concussion. This was conducted to aid in the improvement of sports related policy and regulations to mitigate the health risk brought on by the dangers of concussion. In the development of an automated sentiment analysis system, the performance of varying neural network models including: FFNN, CNN, RNN, GRU, LSTM, and TSN, was evaluated. The evaluation was performed to investigate the optimal set of neural network models that can be used in an ensemble arrangement to predict the superior sentiment between the risk of concussion and the winning culture mentality within sports. The evaluation yielded the following optimal models for a fully connect neural network ensemble system:

- non-pre-trained single layer CNN with filter sizes [1, 2],
- non-pre-trained single layer CNN with filter sizes [1, 2, 3],
- non-pre-trained multi-layer CNN,
- non-pre-trained LSTM,
- single layer CNN pre-trained on SemEval-2016 with filter sizes [1, 2, 3],
- multi-layer CNN pre-trained on SemEval-2016, and
- LSTM pre-trained on SemEval-2016.

Lastly, the ensemble system was further trained and yielded a final F1 performance score of 62.71%.

5.1 Contribution

This thesis has made an incremental contribution to the ongoing research and investigation of ensemble based neural network systems for social medial analysis. In summary, the main contributions of this work which relates to the objects stated in section 1.5 are as follows:

1. **Sentiment Pre-processing Pipeline**: The main contribution of this work is the preprocessing pipeline that allows interchangeability between pre-trained word embedding. While the components included in the pre-processing pipeline are standard to sentiment analysis and NLP, the combination of various embedding and the stacking of various components are unique to this body of work. The combination of concatenating pre-trained word vectors and word padding/truncation in the final block of the preprocessing pipeline, provides a method with two main benefits. The concatenation allows varying linguistic and sentiment features to be combined, while the inclusion of word padding/truncation allows a unified document (i.e., tweet) vectorization which can be feed into varying neural network architecture.

2. **Automated Sentiment Analysis System (ASAS) for Sport Related Concussion (SRC) text:** The second contribution is the empirical evaluation of deep neural network in the application of SA of sports related concussion data. The results indicate that various neural network architecture can be employed in the domain of sports related concussion data. While prior works have demonstrated the SA of twitter data, this is the first body of work that attempts to label the sentiment within the domain of sports related concussion. The application to this specific domain introduces a novelty within the SA that has not yet be explored. Prior SA, focus on the SA of a target entity, which attempts to determine the

sentiment of a specific subject within a document. However, the problem presented with SRC, contains two main entities (concussion and sports) which are being ranked among each other. This introduces some additional technical challenges not present in the standard target-level sentiment analysis. Since two entities are ranked, the magnitude of their sentiments must be learned, such that they may be compared. Specifically, in the case of this thesis, the sentiment of concussion is compared to the winning culture of sports.

3. **A methodology for adaptive development of a sentiment analysis system:** The evaluation of the various deep neural network models with the preprocessing pipeline on varying external datasets, indicates that similar methodology can be implemented in the automated analysis system. While no model outperformed the best state-of-the-art model (SemEval-2016 and Senti-Target dataset systems), the results of the models only deviate 2.5% from other comparable state-of-the-art systems. As such, this indicates that the methodology can be applied to other automated analysis system and generate comparable results. However, the potential of the methodology lies within the interchangeability of leveraging different pre-trained embedding's and the flexibility of evaluating various neural network model with minor to no preprocessing changes.

5.2 Direction of Future Works

Based on the work presented in this thesis, the following future work is recommended:

- The understanding of the opinion of the general public with regard to sports related concussion can aid in pushing more emphasis on the need for better policy and regulation to mitigate the risk. However, more information can be gathered to aid in policy and

regulation, if the specific opinions of general public can be also categorized into fine-grained categories. For example, it would be more beneficial to understand how the general public opinion differentiate between the medical treatment conducted to sports related concussion vs the level of education towards concussion in sports. Extracting this understanding could lead the development of policy and regulation towards the specific category requiring attention. As such, a potential direction for future work, is the application of the proposed methodology within this work in the classification of the SRC tweets into fine-grained categories. The work presented in [53] presents the following five important themes in classifying SRC tweets: Medical, Instances of Injury, Education, Policy and Rules, and Subjective Opinion. A challenge portion of this work, is the limited dataset available for analysis of all five categories. The additional levels of classification would require a larger sample size than this body of work. In addition, the growth of the dataset would need to be performed in a way that the sampling among the five categories are balanced, insuring equal representation of each category.

- While the current body of work is conducted on 3-levels of sentiment analysis. An empirical analysis on the methodology of the preprocessing pipeline and the varying neural network architectures can be performed on a finer-grain of SA, evaluating the applicability of the methodology towards 5-levels of SA. The expansion of 5-levels, allows additional sentiments that fall within the gray area between positive, neutral, and negative. This could remove the ambiguity of documents which are not clearly ‘positive’, ‘neutral’ or ‘negative’ For instance, ‘somewhat positive’ can be utilized to categorize documents which are between ‘positive’ and ‘neutral’. Rather than determining, which is more prominent

('positive' or 'neutral'), the document can be labelled as 'somewhat positive'. Similarly, 'somewhat negative' can be used in a similar way.

- Extension to the preprocessing pipeline could be explored to incorporate target-topic features for target-level sentiment analysis.

Appendices

Appendix A

This appendix provides a central location for all environment parameter's used in the implementation of this thesis's current body of works. It outlines, the programing environment, dependent packages and toolkits that were used within the development of the system. The following summaries the implementation configuration:

- **Language:**
    - Python: v2.7 (environment for neural network system)
    - Python: v3.5 (environment for preprocessing pipeline)
- **Python 2.7 Packages:**
    - jumpy: v1.12 (used as main data structure to manipulate vector and matrix data)
    - pandas: v0.22 (used in primarily to quick load file data and convert to jumpy)
    - matplotlib: v2.2 (used to graph evaluation and results during experiment)
    - scikit-learn: v0.19 (used to calculate evaluation metrics)
    - keras: v2.1 (used as high-level api build neural networks)
    - tensorflow: v1.1 (used as primary network library below keras)
- **Python 3.5 Packages:**
    - nltk: v3.3 (natural language tool kit to preprocess tweets)
    - genism: v3.4 (used to load pre-trained word vectorization models within preprocessing pipeline)
    - stanfordnlp: v3.8 (used as a wrapper to connect with Stanford core nlp system)
    - ekphrasis: v0.4 (additional text processing tool to perform word segmentation [51])
    - cython: v0.26 (used as an additional package to speed-up ekpharsis libraries)
    - numpy: v1.14
- **External Toolkits:**
    - numpy: v1.14
    - Stanford's CoreNLP: v3.9 [68]

Appendix B

This appendix provides a list of part-of-speech tags in the Penn Treebank that is utilized in this current body of works.

INDEX	PART-OF-SPEECH TAG	DESCRIPTION
01	*CC*	*Coordinating conjunction*
02	*CD*	*Cardinal number*
03	*DT*	*Determiner*
04	*EX*	*Existential there*
05	*FW*	*Foreign word*
06	*IN*	*Preposition or subordinating conjunction*
07	*JJ*	*Adjective*
08	*JJR*	*Adjective, comparative*
09	*JJS*	*Adjective, superlative*
10	*LS*	*List item marker*
11	*MD*	*Modal*
12	*NN*	*Noun, singular or mass*
13	*NNS*	*Noun, plural*
14	*NNP*	*Proper noun, singular*
15	*NNPS*	*Proper noun, plural*
16	*PDT*	*Pre-determiner*
17	*POS*	*Possessive ending*
18	*PRP*	*Personal pronoun*
19	*PRP$*	*Possessive pronoun*
20	*RB*	*Adverb*
21	*RBR*	*Adverb, comparative*
22	*RBS*	*Adverb, superlative*
23	*RP*	*Particle*
24	*SYM*	*Symbol*

25	*TO*	*To*
26	*UH*	*Interjection*
27	*VB*	*Verb, base form*
28	*VBD*	*Verb, past tense*
29	*VBG*	*Verb, gerund or present participle*
30	*VBN*	*Verb, past participle*
31	*VBP*	*Verb, non-3rd person singular present*
32	*VBZ*	*Verb, 3rd person singular present*
33	*WDT*	*Wh-determiner*
34	*WP*	*Wh-pronoun*
35	*WP$*	*Possessive wh-pronoun*
36	*WRB*	*Wh-adverb*

Appendix C

This appendix provides additional results for the neural network models and the different datasets.

Precision

MODEL	LAYER/FILTER SIZES	SRC	SEM-EVAL	KG	RT	SENTI-TARGET	UCI	UMICH 650
FFNN	[400, 400]	0.5508	0.5931	0.9044	0.8278	0.5762	0.7932	0.8707
	[400, 400, 400]	0.5545	0.5891	0.9054	0.8286	0.5755	0.7898	0.8772
	[775, 225, 75, 25]	0.5499	0.5705	0.9089	0.8270	0.5680	-	-
	[400, 200, 100, 50]	-	-	-	-	-	0.8029	0.8698
SINGLE LAYER CNN	[1, 2]	0.6234	0.6329	0.9325	0.8722	0.6145	0.8508	0.8984
	[1, 2, 3]	0.6213	0.6271	0.9351	0.8786	0.6264	0.8496	0.8901
	[3, 4, 5]	0.5973	0.6256	0.9281	0.8693	0.6252	0.8368	0.8937
	[1, 2, 3, 4, 5]	0.6137	0.6159	0.9323	0.8736	0.6214	0.8489	0.9041
MULTI-LAYER CNN	[1, 2]	0.6200	0.6230	0.9320	0.8730	0.6086	0.8546	0.8937
GRU	[205]	0.6187	0.6119	0.9329	0.8677	0.5973	0.8473	0.9004
LSTM	[205]	0.6166	0.6166	0.9325	0.8717	0.6059	0.8520	0.8966
BI-DIR LSTM	[205]	0.6084	0.6201	0.9328	0.8726	0.6156	0.8467	0.8804
TCN	[3]	0.5595	0.5965	0.9294	0.8587	0.5935	0.8431	0.8997

Recall

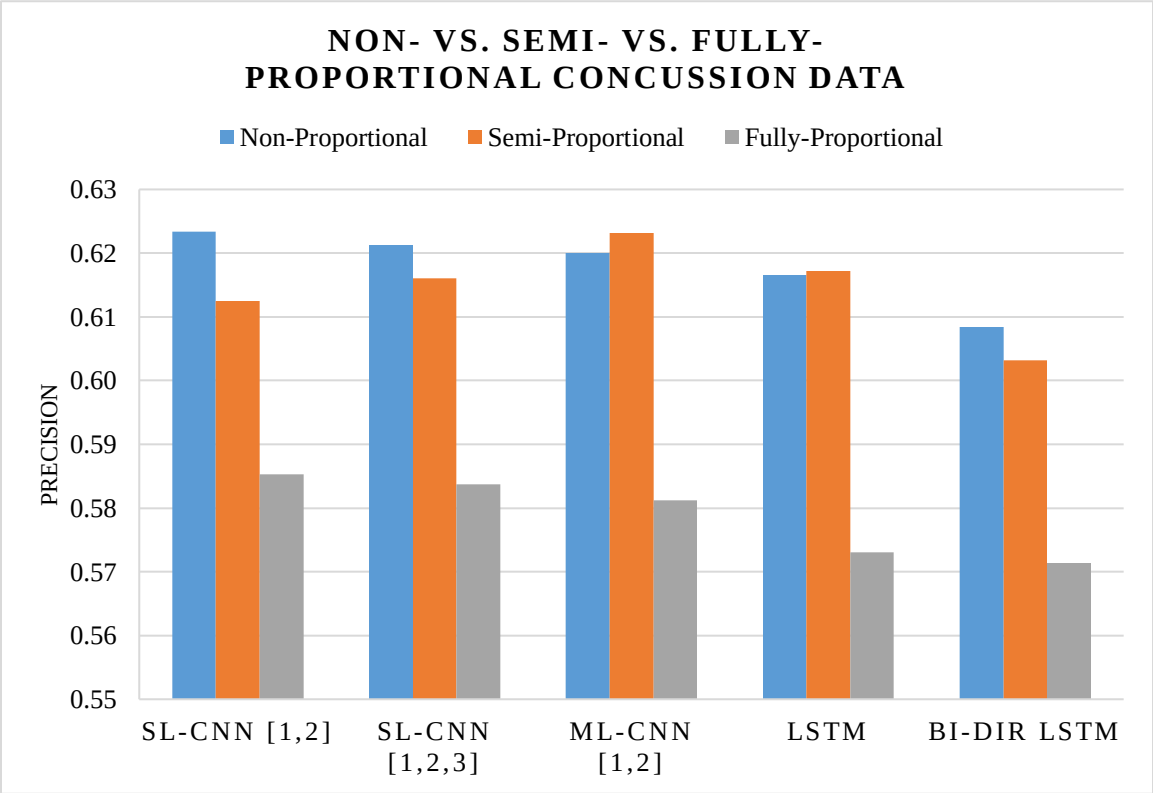
MODEL	LAYER/FILTER SIZES	SRC	SEM-EVAL	KG	RT	SENTI-TARGET	UCI	UMICH 650
FFNN	[400, 400]	0.5649	0.5950	0.9045	0.8373	0.5788	0.7904	0.8689
	[400, 400, 400]	0.5649	0.5899	0.9054	0.8338	0.5772	0.7887	0.8689
	[775, 225, 75, 25]	0.5582	0.5884	0.9066	0.8210	0.5699	-	-
	[400, 200, 100, 50]	-	-	-	-	-	0.8024	0.8689
SINGLE LAYER CNN	[1, 2]	0.6184	0.6174	0.9324	0.8761	0.6168	0.8505	0.8932
	[1, 2, 3]	0.6101	0.6108	0.9344	0.8760	0.6265	0.8454	0.8883
	[3, 4, 5]	0.6057	0.6021	0.9282	0.8736	0.6257	0.8368	0.8932
	[1, 2, 3, 4, 5]	0.6117	0.6062	0.9324	0.8761	0.6233	0.8488	0.9029
MULTI-LAYER CNN	[1, 2]	0.6123	0.6220	0.9319	0.8753	0.6103	0.8505	0.8932
GRU	[205]	0.6035	0.6057	0.9324	0.8655	0.5974	0.8471	0.8932
LSTM	[205]	0.6117	0.6118	0.9326	0.8721	0.6063	0.8505	0.8883
BI-DIR LSTM	[205]	0.6073	0.6215	0.9328	0.8734	0.6176	0.8419	0.8786
TCN	[3]	0.5595	0.5960	0.9289	0.8536	0.5950	0.8402	0.8981

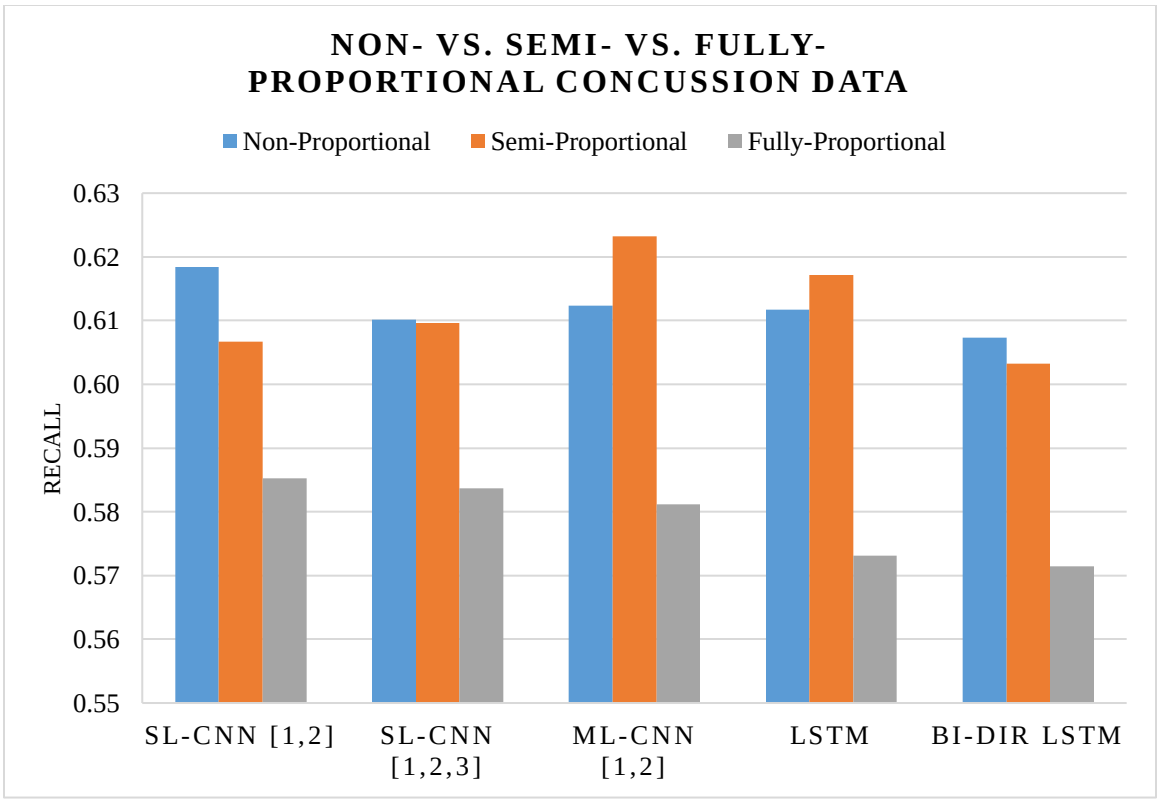
Accuracy

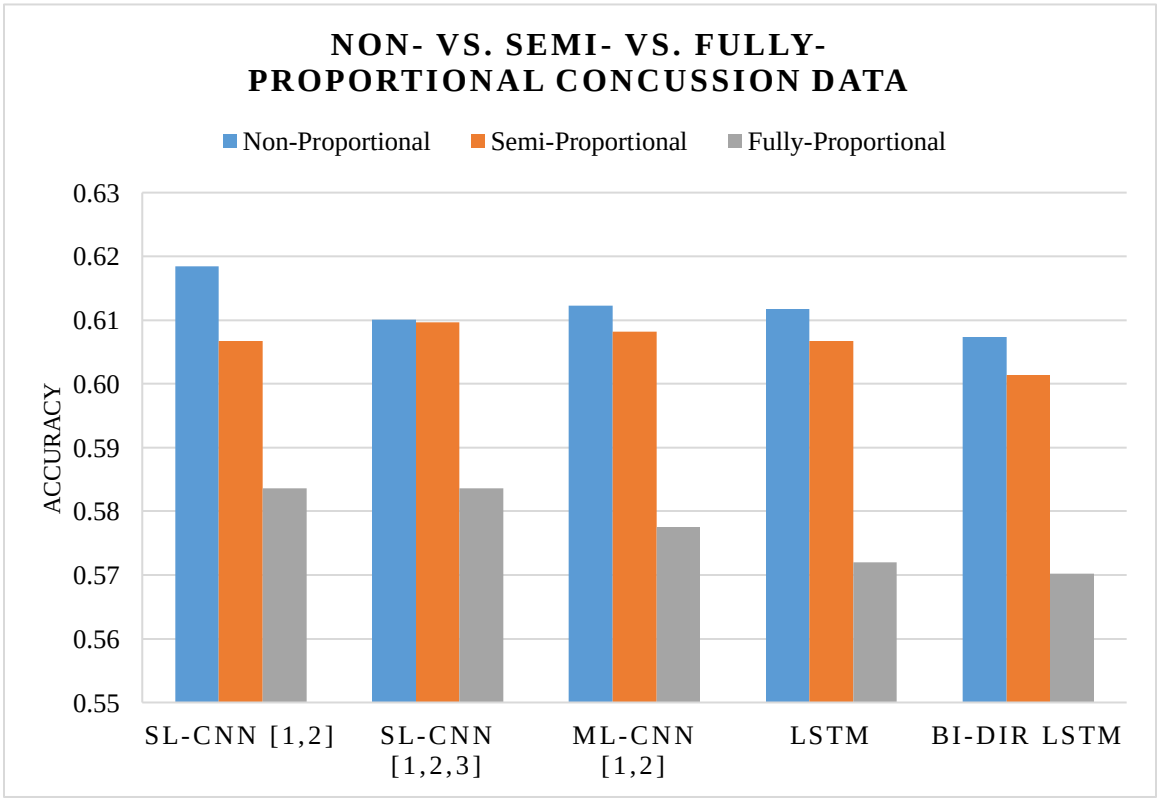
MODEL	LAYER/FILTER SIZES	SRC	SEM-EVAL	KG	RT	SENTI-TARGET	UCI	UMICH 650
FFNN	[400, 400]	0.5649	0.5950	0.9045	0.8373	0.5788	0.7904	0.8689
	[400, 400, 400]	0.5652	0.5899	0.9054	0.8338	0.5772	0.7887	0.8689
	[775, 225, 75, 25]	0.5582	0.5884	0.9066	0.8210	0.5699	-	-
	[400, 200, 100, 50]	-	-	-	-	-	0.8024	0.8689
SINGLE LAYER CNN	[1, 2]	0.6184	0.6174	0.9324	0.8761	0.6168	0.8505	0.8932
	[1, 2, 3]	0.6101	0.6108	0.9344	0.8760	0.6265	0.8454	0.8883
	[3, 4, 5]	0.6057	0.6021	0.9282	0.8736	0.6257	0.8368	0.8932
	[1, 2, 3, 4, 5]	0.6117	0.6062	0.9324	0.8772	0.6233	0.8488	0.9029
MULTI-LAYER CNN	[1, 2]	0.6123	0.6220	0.9319	0.6103	0.6103	0.8505	0.8932
GRU	[205]	0.6035	0.6057	0.9324	0.8655	0.5974	0.8471	0.8932
LSTM	[205]	0.6117	0.6118	0.9326	0.8721	0.6063	0.8505	0.8883
BI-DIR LSTM	[205]	0.6073	0.6215	0.9328	0.8734	0.6176	0.8419	0.8786
TCN	[3]	0.5595	0.5960	0.9289	0.8536	0.5950	0.8402	0.8981

Appendix D

This appendix provides additional graphs illustrating the performance of the different proportional concussion datasets using the other evaluation metrics.




NON- VS. SEMI- VS. FULLY-PROPORTIONAL CONCUSSION DATA
Non-Proportional
Semi-Proportional
Fully-Proportional
RECALL
0.63
0.62
0.61
0.60
0.59
0.58
0.57
0.56
0.55
SL-CNN [1,2]
SL-CNN [1,2,3]
ML-CNN [1,2]
LSTM
BI-DIR LSTM


NON- VS. SEMI- VS. FULLY-PROPORTIONAL CONCUSSION DATA
Non-Proportional
Semi-Proportional
Fully-Proportional
ACCURACY
0.63
0.62
0.61
0.60
0.59
0.58
0.57
0.56
0.55
SL-CNN [1,2]
SL-CNN [1,2,3]
ML-CNN [1,2]
LSTM
BI-DIR LSTM

References

[1] C. Heritage and C. Heritage, "Concussions in sport," *aem*, 01-Nov-2017. [Online]. Available: https://www.canada.ca/en/canadian-heritage/services/concussions.html. [Accessed: 28-Aug-2018].

[2] "TBI: Get the Facts | Concussion | Traumatic Brain Injury | CDC Injury Center," 16-Oct-2017. [Online]. Available: https://www.cdc.gov/traumaticbraininjury/get_the_facts.html. [Accessed: 22-May-2018].

[3] US Department of Health & Human Services; Centers for Disease Control (CDC); National Center for Injury Prevention and Control, "Report to Congress on Mild Traumatic Brain Injury in the United States: Steps to Prevent a Serious Public Health Problem: (371602004-001)." American Psychological Association, 2003.

[4] S. B. Johnson, R. W. Blum, and J. N. Giedd, "Adolescent Maturity and the Brain: The Promise and Pitfalls of Neuroscience Research in Adolescent Health Policy," *J. Adolesc. Health Off. Publ. Soc. Adolesc. Med.*, vol. 45, no. 3, pp. 216–221, Sep. 2009.

[5] CDC, "Benefits of Physical Activity," *Centers for Disease Control and Prevention*, 13-Feb-2018. [Online]. Available: https://www.cdc.gov/physicalactivity/basics/pa-health/index.htm. [Accessed: 22-May-2018].

[6] P. Burstein, "The Impact of Public Opinion on Public Policy: A Review and an Agenda," *Polit. Res. Q.*, vol. 56, no. 1, pp. 29–40, 2003.

[7] B. Bars *et al.*, "Opportunities For Change In The Current Nba 'One And Done' DraftSystem," *Pa. State Univ. Pres. Leadersh. Acad.*, p. 22, May 2018.

[8] J. M. Kennedy and B. Vargus, "Challenges in Survey Research and Their Implications for Philanthropic Studies Research," *Nonprofit Volunt. Sect. Q.*, vol. 30, no. 3, pp. 483–494, Sep. 2001.

[9] "Collecting survey data," *Pew Research Center*, 29-Jan-2015. [Online]. Available: http://www.pewresearch.org/methodology/u-s-survey-research/collecting-survey-data/. [Accessed: 22-May-2018].

[10] "Twitter MAU worldwide 2018 | Statistic," *Statista*. [Online]. Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/. [Accessed: 22-May-2018].

[11] R. Feldman, "Techniques and Applications for Sentiment Analysis," *Commun ACM*, vol. 56, no. 4, pp. 82–89, Apr. 2013.

[12] J. Kajornrit and P. Chaipornkaew, "A comparative study of ensemble back-propagation neural network for the regression problems," in *2017 2nd International Conference on Information Technology (INCIT)*, 2017, pp. 1–6.

[13] N. Jindal and B. Liu, “Identifying Comparative Sentences in Text Documents,” in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2006, pp. 244–251.

[14] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[15] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[16] I. Goodfellow, Y. Bengio, and A. Courville, “Regularization for Deep Learning,” in *Deep Learning*, MIT Press, 2016, pp. 224–270.

[17] I. Goodfellow, Y. Bengio, and A. Courville, “Optimization for Training Deep Models,” in *Deep Learning*, MIT Press, 2016, pp. 271–325.

[18] B. Ding, H. Qian, and J. Zhou, “Activation functions and their characteristics in deep neural networks,” in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1836–1841.

[19] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[20] P. Munro, “Backpropagation,” in *Encyclopedia of Machine Learning and Data Mining*, Springer, Boston, MA, 2017, pp. 93–97.

[21] A. W. Trask, “Gradient Descent,” in *Grokking Deep Learning*, Manning Publications, 2018, pp. 47–77.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] “Loss Functions — ML Cheatsheet documentation.” [Online]. Available: http://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html. [Accessed: 24-May-2018].

[24] D. Campbell, R. A. Dunne, and N. A. Campbell, *On The Pairing Of The Softmax Activation And Cross–Entropy Penalty Functions And The Derivation Of The Softmax Activation Function*. .

[25] W. Cavnar and J. M. Trenkle, “N-Gram-Based Text Categorization,” in *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.

[26] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A Convolutional Neural Network for Modelling Sentences,” *ArXiv14042188 Cs*, Apr. 2014.

[27] “Understanding LSTM Networks -- colah’s blog.” [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 25-May-2018].

[28] D. Britz, “Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs,” *WildML*, 17-Sep-2015. [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/. [Accessed: 17-Jul-2018].

[29] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[30] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[31] S. Hochreiter, “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions,” *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 06, no. 02, pp. 107–116, Apr. 1998.

[32] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, vol. 9. 1997.

[33] “How RNNs and LSTM Work?” [Online]. Available: https://elham-khanche.github.io/blog/RNNs_and_LSTM/. [Accessed: 25-May-2018].

[34] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” *ArXiv14061078 Cs Stat*, Jun. 2014.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *ArXiv14123555 Cs*, Dec. 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015.

[37] S. Bai, J. Z. Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,” *ArXiv180301271 Cs*, Mar. 2018.

[38] F. Yu and V. Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions,” *ArXiv151107122 Cs*, Nov. 2015.

[39] S. M. Mohammad and P. D. Turney, “Nrc emotion lexicon,” *Natl. Res. Counc. Can.*, 2013.

[40] P. D. Turney and M. L. Littman, “Measuring Praise and Criticism: Inference of Semantic Orientation from Association,” *ACM Trans Inf Syst*, vol. 21, no. 4, pp. 315–346, Oct. 2003.

[41] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” in *in Proc. of LREC*, 2010.

[42] S. Kiritchenko, X. Zhu, and S. M. Mohammad, “Sentiment analysis of short informal texts,” *J. Artif. Intell. Res.*, vol. 50, pp. 723–762, 2014.

[43] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-Based Methods for Sentiment Analysis,” *Comput. Linguist.*, vol. 37, no. 2, pp. 267–307, Jun. 2011.

[44] P. Chaovalit and L. Zhou, "Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005, pp. 112c-112c.

[45] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters," *Comput. Intell. 0824-7935*, vol. 22, no. 2, pp. 110–125, 2006.

[46] A. Balahur *et al.*, "Sentiment Analysis in the News," *ArXiv13096202 Cs*, Sep. 2013.

[47] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov, "SemEval-2015 Task 10: Sentiment Analysis in Twitter," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 451–463.

[48] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "SemEval-2016 task 4: Sentiment analysis in Twitter," in *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, 2016, pp. 1–18.

[49] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," presented at the Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 502–518.

[50] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, and M. Jaggi, "SwissCheese at SemEval-2016 Task 4: Sentiment Classification Using an Ensemble of Convolutional Neural Networks with Distant Supervision," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, San Diego, California, 2016, pp. 1124–1128.

[51] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, 2017, pp. 747–754.

[52] M. Cliche, "BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs," *ArXiv170406125 Cs Stat*, Apr. 2017.

[53] A. M. Workewych, M. Ciuffetelli Muzzi, R. Jing, S. Zhang, J. Topolovec-Vranic, and M. D. Cusimano, "Twitter and traumatic brain injury: A content and sentiment analysis of tweets pertaining to sport-related brain injury," *SAGE Open Med.*, vol. 5, p. 2050312117720057, 2017.

[54] "Twitter Developer Platform." [Online]. Available: https://developer.twitter.com/content/developer-twitter/en.html. [Accessed: 02-Jul-2018].

[55] seirasto, *Semeval Twitter data download script + user info*. 2018.

[56] "Partly Sunny with a Chance of Hashtags." [Online]. Available: https://www.kaggle.com/c/crowdflower-weather-twitter. [Accessed: 02-Jul-2018].

[57] “Sentiment Analysis on Movie Reviews.” [Online]. Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews. [Accessed: 02-Jul-2018].

[58] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, “Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland, 2014, pp. 49–54.

[59] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, “From group to individual labels using deep features,” presented at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 597–606.

[60] “UCI Machine Learning Repository: Sentiment Labelled Sentences Data Set.” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences. [Accessed: 07-Aug-2018].

[61] “UMICH SI650 - Sentiment Classification.” [Online]. Available: https://www.kaggle.com/c/si650winter11. [Accessed: 02-Jul-2018].

[62] “Stemming and lemmatization.” [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html. [Accessed: 23-May-2018].

[63] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by gibbs sampling,” presented at the Proceedings of the 43rd annual meeting on association for computational linguistics, 2005, pp. 363–370.

[64] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” presented at the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[65] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, “Sentiment Analysis of Twitter Data,” in *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, Portland, Oregon, 2011, pp. 30–38.

[66] B. Santorini, “Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision),” *Tech. Rep. CIS*, p. 37, 1990.

[67] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Dec. 2014.

[68] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP natural language processing toolkit,” presented at the Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 55–60.